NAVY OCCUPATIONAL HEALTH
INFORMATION MANAGEMENT SYSTEM

NOHIMS

(1)

SYSTEM-WIDE MODULE
PROGRAM MAINTENANCE MANUAL

JUNE 1987

Prepared by
The MITRE Corporation
McLEAN, Virginia

REPORT DOCUMENTATION PAGE	1. REPORT NO. DOD/SW/MT-88/017c	2.	3. Recipient's Accession No.

4. Title and Subtitle

NAVY OCCUPATIONAL HEALTH INFORMATION MANAGEMENT SYSTEM (NOHIMS) SYSTEM-WIDE MODULE PROGRAM MAINTENANCE MANUAL

5. Report Date JUNE 1987

6.

7. Author(s)

MITRE CORPORATION

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

NAVY REGIONAL DATA AUTOMATION CENTER
CODE 4443
WASHINGTON NAVY YARD
WASHINGTON, DC 20374

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.

(C)

(G)

12. Sponsoring Organization Name and Address

NAVY ENVIRONMENTAL HEALTH CENTER
NAVAL STATION
NORFOLK, VA 23511-6695
ATTN: CODE 40/ CDR JAMES ALLEN

13. Type of Report & Period Covered

14.

15. Supplementary Notes

For magnetic tape, see: ADA198157

16. Abstract (Limit: 200 words)

THE SYSTEM-WIDE PROGRAM MAINTENANCE MANUAL DESCRIBES THE SOFTWARE USED IN THE ADMINISTRATION MODULE AND THOSE SOFTWARE UTILITIES THAT ARE USED BY ALL MODULES. THIS DOCUMENT IS INTENDED FOR USE BY THE MAINTENANCE PROGRAMMER.

17. Document Analysis a. Descriptors

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement

RELEASE UNLIMITED

19. Security Class (This Report) UNCLASSIFIED

20. Security Class (This Page) UNCLASSIFIED

21. No. of Pages 150

22. Price

Naval Sea Systems Command
Occupational Safety and Health
Record Keeping System

System-Wide Module
Program Maintenance Manual

June 1987

Prepared by
The MITRE Corporation
McLean, Virginia

PREFACE

Since August 1984, The MITRE Corporation has been supporting the Naval Sea Systems Command (NAVSEA) and the Naval Medical Command (NAVMEDCOM) in their joint efforts to enhance the Navy Occupational Health Information Management System (NOHIMS). NOHIMS, whose initial version was developed at the Naval Health Research Center (NHRC), is a composite of two subsystems: an industrial subsystem and a medical subsystem. The goal of the enhancement effort was to create a comprehensive occupational health and safety system for Navy industrial facilities by expanding upon the original NOHIMS functions and adding modules for hazard deficiency abatement, hazardous material control, injury claims and compensation, and safety and health training. To meet this goal, MITRE developed an enhanced industrial subsystem, referred to as the Occupational Safety and Health Record Keeping System (OSHRKS), using a prototyping approach and a public domain data base management software package, the Veterans Administration's (VA's) FileManager (FileMan).

OSHRKS consists of the following seven modules:

- Environmental Exposure
- Medical Exam Scheduling
- Hazardous Materials Control
- Hazard Deficiency Abatement
- Injury and Compensation Claims
- Safety and Health Training
- Administration

Each NAVSEA facility will use from four to seven of these modules depending on its information needs. The NAVMEDCOM sites require three of the industrial modules and the Administration module in addition to the medical subsystem.

Complete and accurate technical and non-technical documentation was required for each of these modules. This documentation was to describe clearly and accurately the capabilities of OSHRKS--an advanced, online, integrated system based on the use of a data base management system and a programmer tool kit--while also satisfying the Navy's documentation standards. Representatives from various groups within the Navy, working

with members of MITRE's technical staff, created a set of documentation guidelines for the OSHRKS modules. These guidelines specified the title of each document and its content and format.

The following three types of documents have been prepared for each of the first six modules listed above:

- Users' Manual - This manual describes, in non-technical terms, the module's major input and output processes. Examples of reports and displays produced by the module are included. This document is intended for use by the reader who is interested in understanding the module's capabilities.

- Operators' Guide - This guide explains how a user interacts with the module to enter or retrieve data. For each menu option in a module, an overview of the purpose of the option is presented, an example prompt sequence is displayed, and detailed explanations of the user's interactions to specific prompts are discussed. These documents are intended for use by those people who will be entering data into or retrieving data from the module.

- Program Maintenance Manual - This manual describes the software used by the module and is intended for use by the programmer who must maintain or enhance the module's software.

Three additional documents that provide documentation on the Administration module and on system-wide activity have also been prepared. The Primer describes, in general, how a user interacts with a FileMan-based system and enters and retrieves data from the Administration module. The System Manager's Guide provides instructions to the staff that must keep the system operational on a day-to-day basis. Largely, it serves as the Operators' Guide for the Administration module. System management functions needed to keep the other modules operational are also explained in this document. The System-Wide Program Maintenance Manual describes the software used in the Administration module and those software utilities that are used by all modules. This document is intended for use by the maintenance programmer.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES

LIST OF FIGURES (Concluded)

LIST OF TABLES

1.0 GENERAL

1.1 Purpose of the Program Maintenance Manual

This manual describes the software used by the Administration (ADMIN) module of the Naval Sea Systems Command's (NAVSEA's) Occupational Safety and Health Record Keeping System (OSHRKS) and the System-Wide utility software. An overview of the ADMIN module is presented, followed by detailed descriptions of the routines that support each function in the module and the utility volumes that are used system-wide. The information in this document is intended to help computer support staff maintain the system and the ADMIN module software. A detailed discussion of how the user interacts with the ADMIN module is found in the System Manager's Guide. Keywords: Management information Systems, Occupational diseases, Health surveys, Medical examination, Military medicine,

1.2 References

The following references provide additional information on the OSHRKS:

- VA FileMan User's Manual, Version 17, Veterans Administration, March 1986 Industrial medicine, Hazardous materials. (SDW)
- VA FileMan Programmer's Manual, Version 17, Veterans Administration, March 1986
- Environmental Exposure Users' Manual
- Environmental Exposure Operators' Guide
- Environmental Exposure Program Maintenance Manual
- Medical Exam Scheduling Users' Manual
- Medical Exam Scheduling Operators' Guide
- Medical Exam Scheduling Program Maintenance Manual
- Hazardous Materials Control Users' Manual
- Hazardous Materials Control Operators' Guide
- Hazardous Materials Control Program Maintenance Manual
- Hazardous Deficiency Abatement Users' Manual
- Hazardous Deficiency Abatement Operators' Guide
- Hazardous Deficiency Abatement Program Maintenance Manual

- Safety and Health Training Users' Manual
- Safety and Health Training Operators' Guide
- Safety and Health Training Program Maintenance Manual
- Injury and Compensation Claims Users' Manual
- Injury and Compensation Claims Operators' Guide
- Injury and Compensation Claims Program Maintenance Manual
- OSHRKS Primer
- System Manager's Guide
- VA Kernel Reference Manual

The following references provide background on Navy occupational safety and health practices:

- Navy Occupational Safety and Health Program Manual, OPNAV Instruction 5100.23B, 31 August 1983
- Shipyard Training Development Management Plan, Naval Sea Systems Command, January 1985
- Department of Defense Hazardous Materials Information System Procedures, Department of Defense 6050.5M July 1981
- Federal Supply Classification Part 1: Group and Classes, SB 708-21, May 1982
- Materials Management Application - Shelf File/Hazardous Materials Analysis Package, August 1984
- Federal Standard - Material Safety Data Sheets, Preparation and Submission of (Proposed), Federal Standards 313C, April 1983
- NAVSUP Instruction 5100.27: Navy Hazardous Material Control Program
- Consolidated Hazardous Item List (CHIL) - NAVSUP Publication 4500, July 1980
- Executive Order 12196
- Code of Federal Regulations Title 29 #1960.30
- NAVSEA Instruction 5100.15

- Department of Defense Occupational Health Surveillance Manual, DoD 6055.5-M

- American National Standard Method of Recording Basic Facts Relating to the Nature and Occurrence of Work Injuries, ANSI Standard Z16.2

- NAVSEA Instruction 12810.1, Employee Occupational Illness/Injury Compensation Claims; Management and Reporting of

- Training for Federal Employing Agency Compensation Specialists Resource Book, Federal Employee's Compensation Program, U.S. Department of Labor, Employee Standards Administration

- OPNAV Instruction 5102.1B, Mishaps Investigation and Reporting

1.3 Terms and Abbreviations

Appendix B to the System Manager's Guide contains a glossary of all the terms used in this document and all referenced OSHRKS Users' Manuals, Operators' Guides, and Program Maintenance Manuals.

1.4 Programming Language(s) and Conventions

The ADMIN module software is written in the Massachusetts General Hospital Utility Multi-Programming System (MUMPS) programming language. MUMPS is a standard language (X11.1-1984) approved by the American National Standards Institute (ANSI), though non-standard dialects exists. Non-standard language features have been avoided as much as possible so that OSHRKS can run in any standard MUMPS environment.

Certain features of OSHRKS, such as error trapping, require the use of implementation-specific language features. When necessary, these features are implemented via M/VX, the InterSystems Corporation's MUMPS language product for the VAX computer. For each option in this manual using non-standard features, an explicit discussion of the feature is provided.

The OSHRKS software is based on the use of two MUMPS-based software packages: the VA FileMan data base management system and the VA Kernel system management packages. Knowledge of FileMan is essential to the maintenance programmer. Extensive use is made of FileMan input templates, sort templates, and print templates. Many OSHRKS options use direct calls to FileMan utility routines, e.g., DIC, DIP, DIQ, DIE, DIWF, within the MUMPS code to perform such activities as lookup, print, inquiry, input, and form letter print, respectively. Additionally, through the use of templates and data dictionaries, certain security features of FileMan are activated in the OSHRKS. Furthermore, ad hoc query in OSHRKS is done through the use of the FileMan Search (Option 3) and Print (Option 2)

options. Also, the FileMan data dictionary* is used to define all of the files in this module. The reader must have carefully reviewed the FileMan User's Manual and the FileMan Programmer's Manual, published by the Veterans Administration (VA), before using this manual or some of the terminology, specific to FileMan, used in this manual will be unclear.

The Kernel package is used in OSHRKS to provide security (user and device levels), menu management, and task management. Where custom MUMPS code has been used, the Kernel sets FileMan variables and invokes a FileMan routine to perform the appropriate function. For a complete technical view of OSHRKS, this manual must be used in conjunction with the documentation provided by the VA on the FileMan and Kernel packages (see Section 1.2).

1.5 Organization of This Report

Section 2.0 provides a non-technical overview of the ADMIN module and describes files maintained in the module. Section 3.0 discusses the ADMIN menu options and provides the number of the section where each option is discussed. Sections 4.0 through 8.0 describe the software used by the various menu options. Each section covers options that perform related functions. Section 9.0 reviews system-wide utility routines that support file editting and major file lookups. Section 10.0 discusses several FileMan features used in the system that are not covered by the FileMan documentation.

Each option's or utility's description in Sections 4.0 through 8.0 contains the following subsections:

- Purpose - Describes in non-technical terms the function(s) which the option performs.
- Overview - Describes the type of option and the templates, files, subfiles, and routines it uses. If the option is a routine option, i.e., it invokes the use of custom MUMPS code, the flow among routines and each routine's major function(s) are described.
- Globals Referenced - Lists by name and file number each file and subfile read or updated, the global referenced, and the module that has ownership of the global.
- Variables - Lists each variable name with a definition of its use.

*If the maintenance programmer using this manual needs to review a file's data dictionary entries, he or she should use the FileMan List File Attributes option to generate the most current file data dictionary.

- **Remarks** - Describes any special processing, special coding conventions, algorithms, interface consequences, triggers, computed fields, and input syntax checks that are specific to the option. If a module-specific utility routine or software feature is involved, the reader is referred to the appropraite section of the manual.

1.6 Routine Structure Diagram Conventions

Most of the options in the ADMIN module are routine options, meaning that the Kernel invokes a custom-coded set of routines to perform the function(s) embodied in the routine. For each of these options and for the system-wide utility routines, a routine structure diagram is included to describe the set of routines that are used. Each routine in the structure diagram is shown as a rectangle. The structure diagram indicates the control flow within the routine by both the positioning of the retangles and the orientation of the connecting lines. When two rectangles are connected by vertical lines without arrows, the upper program is "calling" the lower program through the use of the MUMPS "DO" command. The "called" routine returns control back to the "calling" program when the "called" routine completes its work. When two rectangles are connected by a line with arrows, this means that one routine is passing control to the other by means of a MUMPS "GO TO" command; usually these routines appear horizontally in the diagram. The direction of the arrow indicates the direction in which control is transferred. When a module-wide utility routine is invoked as part of an option, the routine is marked with a single asterisk (*) on the routine structure diagram. The reader must refer to Section 9.0 of this manual for a complete description of the utility routine (and its internal flow). When a system-wide utility routine is used, e.g., T2GED (the standard input driver program that performs completeness and consistency checking), the routine structure diagram shows this routine with a double asterisk (**). System-wide utilities are also discussed in Section 9 of this document.

2.0 SYSTEM DESIGN OVERVIEW

2.1 System Design Summary

From the beginning of OSHRKS's development, it was known that the major files maintained by the system had to be integrated. Though the system was to contain six different modules, the following six files emerged as those which had common use by all six modules:

- Ageney Unit
- Employee
- Location
- Stressor
- Occupation
- Operation

These files are used by the system to control the values that can be entered for such data fields as Agency Units, Employees, Locations, Occupations, Operations, and Stressors. When used this way, these files function as reference files. However, three of these files--Agency Unit, Employee, and Stressor--also contain considerable data in each file entry and, hence, also function as applications files. For instance, the Employee file stores the person's name, social security number, address, shop assignment, hire date(s), and similar personnel-related data. Whenever these fields appear on a report, regardless of which module produces the report, the data is retrieved from the Employee file. This provides the integrated data base capability that the system requires. As the system was developed, the software used to create and maintain this set of files became a seventh module, known as the Administration (ADMIN) module. An overview of the six major files in the ADMIN Module and the sixteen files that support them is presented in Section 2.2.

2.2 File Overview

The 22 files listed in Table 2-1 are all created, read, or updated in the ADMIN module. Although owned by the MES module, the Clinic file is referenced during processing of the Agency Unit file in the Administration module and thus is considered a supporting file.

TABLE 2-1
FILES AND GLOBALS USED BY
SYSTEM-WIDE MODULE

FILE NAME	FILE NUMBER	GLOBAL REFERENCE	MODULE
Employee	1004	↑EMPLOY(	ADMIN
Agency Unit	1074	↑AGENCY(0,	ADMIN
Location	1073	↑AGENCY(1073,	ADMIN
Operation	1087	↑DIZ(1087,	ADMIN
Occupation	1001	↑DIZ(1001,	ADMIN
Stressor	1083	↑STRESS(0,	ADMIN
State	1003	↑DIZ(1003,	ADMIN
Supervisory Level	1135	↑AGENCY(1047,	ADMIN
NCPDS Transactions	2000	↑PTRANS(	ADMIN
NCPDS Edits	2001	↑DIZ(2001,	ADMIN
Organizational Level	1047	↑AGENCY(1041,	ADMIN
Site	1041	↑DIZ(1041,	ADMIN
Operation Class	1152	↑DIZ(1152,	ADMIN
Operation Subclass	1153	↑DIZ(1153,	ADMIN
Stressor Class	1039	↑STRESS(1039,	ADMIN
Counters	----	↑AGENCY(0,"COUNT,"	ADMIN
Sample Units	1101	↑EXP(1101,	ADMIN
Medical Program	1088	↑MED(1088,	MES
Reason for Medical Visit/ Exam	1128	↑MED(1128,	MES
Clinic	1138	↑MED(0,	MES
Course Catalog	1113	↑TRN(1113,	S/HT
Sampling Goals	1117	↑EXP(1117,	EE

The Medical Program file and Reason for Medical Visit/Exam file are owned by the MES module and are referenced by ADMIN module options and utilities. Similarly, the Course Catalog file is owned by the S/HT module and is used by the Occupation Training Requirements utility during Employee file updates. The Sampling Goals file is maintained/owned by the EE module, but is referenced during the Inactivate Location option.

With these exceptions, the files listed are owned by the ADMIN module although they are often used by other modules. The relationships between the ADMIN module files are illustrated in Figure 2-1.

The Agency Unit file contains the description of the organizational structure of each agency that is part of the OSHRKS data base. Normally, a shipyard will be a single agency. Each department, division, branch, section, and shop is a separate file entry. Entries cannot be deleted. The ID field is Code/Abbreviation. Lookups can also be done on the Name, Level Code, UIC, and Site of Agency Unit fields. The file has a screen which limits user access to specified agencies.

The Location file contains all locations, stored in a four-tiered format: site, location, sublocation, and area. Locations can be inactivated, but not deleted. The ID field is Name. The Date Effective is an identifier. Lookups are done by a special lookup utility.

The Operation file contains all operations with their codes, classes, and subclasses. The file is supplied by NEHC. Entries cannot be deleted. The ID field is Code. Lookups can also be done on the Name, Subclass, and Class fields.

The Stressor file contains all stressors with identifying data, medical monitoring recommendations, exposure limits, acute and chronic effects, and first-aid instructions. Entries cannot be deleted. The ID field is Primary Name. Lookups are done using a special utility which accesses any of the fields described hereafter. The NIOSH number and CAS Number have regular cross references. There is a Short Name that is a mnemonic and a Stripped Name which contains a version of the Primary Name with punctuation and numbers removed. A multiple contains Synonyms and Stripped Syn fields. Another multiple contains a Classification field.

The Employee file contains all employees with identifying data, including terminated, prehire, and current employees. It is a reference file for most of the major module files. Entries cannot be deleted. The ID field is Name. The Internal Name field contains a stripped version of the Name with punctuation and extraneous spaces removed. This field is used as a mnemonic during file lookups. The SSN and Badge Check Number fields are also used during lookups done by the special lookup utility. This file has a screen which limits access to Employee entries based on user agency access.

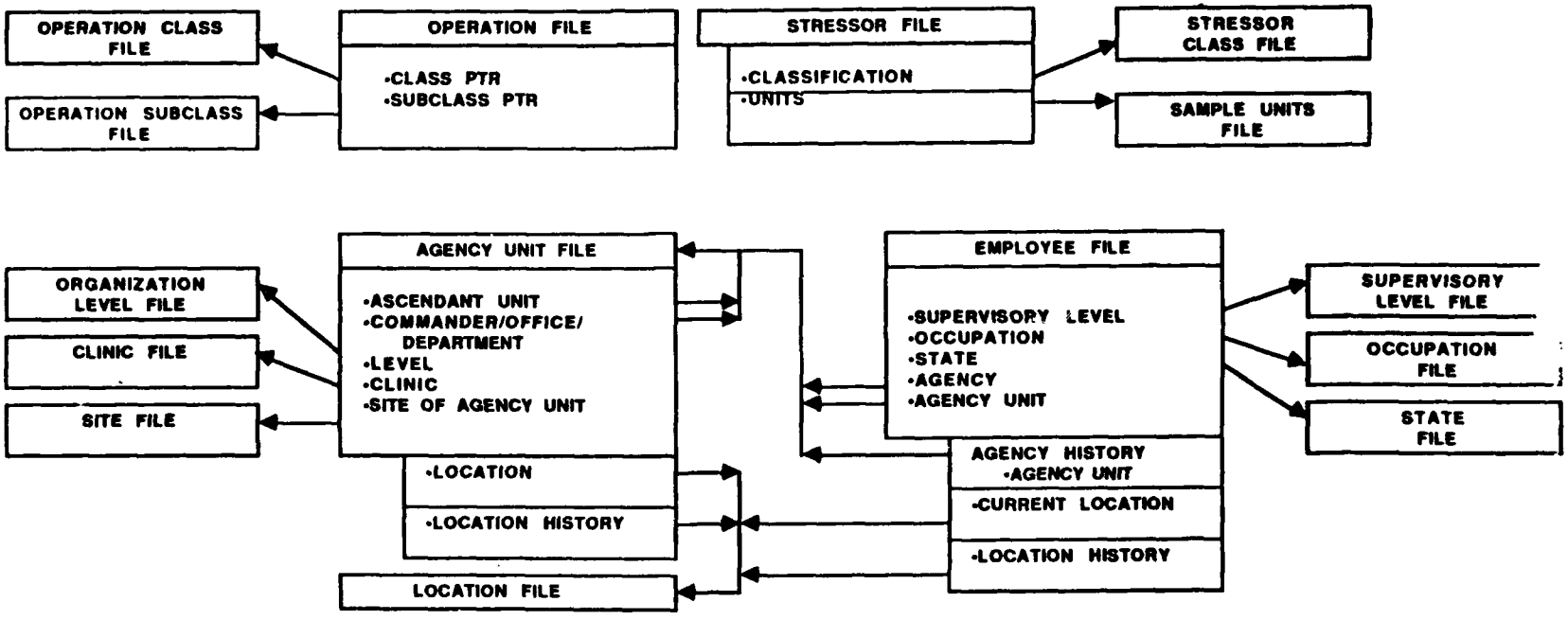

FIGURE 2-1
ADMINISTRATION MODULE FILES

The Occupation file contains civilian Government occupation codes and titles supplied by NEHC. Entries cannot be deleted. The ID field is Occupation Code. Lookups can also be done on the Occupation Title field.

The State file contains standard names and abbreviations as defined nationally. The user does not have access to this file other than to look up entries by the Name and Abbreviation fields.

The Supervisory Level file contains nationally defined names and codes. The user does not have access to this file other than to lookup entries by Name and Code fields.

The NCPDS Transactions file contains tape transactions for update of the Employee file. The ID field is Transaction Number, which is also the file entry number.

The NCPDS Edits file contains data needed to edit the NCPDS transactions file and update the Employee file. The user does not have access to this file.

The Organization Level file contains the levels defined for use in describing an agency structure. Entries cannot be deleted once used in an Agency Unit file entry. The ID field is Name. The Code field is an identifier that can also be used during lookups.

The Site file defines the major sites used by agency units. Entries cannot be deleted once used in a Location description or linked to an Agency Unit. The ID field is Name. File lookups can also use the Abbreviation field.

The Operation Class and Operation Subclass files support the Operation file. Neither file allows entry deletion. Both have only an ID field called Name.

The Stressor Class file supports the Stressor file. Entries are nationally defined. The sole field is the ID field Name.

The Counters file is not a FileMan file. It is a single node containing the alpha characters used to created the Astring field when a new agency is entered.

The Sample Units file supports the Stressor file. It contains a nationally defined table of units of measure. The only field is the ID field Name.

2.3 Naming Conventions

All options names for the ADMIN module in the Kernel begin with "T2" as do all module routines. In general, the third character in the routine names and option names fit the following conventions:

- G: Processes supporting files or performs a utility function
- J: Processes the Agency Unit file
- P: Processes the Employee file
- S: Processes the Stressor file

Variables used in the ADMIN module or system-wide utilities generally start with one of the letters described above to indicate that they support the processing of the corresponding file. In addition, ADMIN module routines will sometimes use variables that fit module conventions as described in the respective module Program Maintenance Manuals.

In addition to their basic definition, several variables are described as being a "corresponding" file or subfile entry number. In such cases, the internal ID field value for the file or subfile is also the entry number in the file or subfile. This was accomplished by including, in the ID field input syntax check, MUMPS code which sets the FileMan variable DINUM equal to the field value, often a pointer. This method helps ensure entry uniqueness and makes processing more efficient by eliminating the need for retrieval of the entry node to get the ID field value.

3.0 MODULE MENUS

The Administration module options listed in Table 3-1 are organized according to the files or modules they support. The options under the Module Tables and Files menu are included in this module menu because they will be used primarily by the System Manager. However, the details on these options are discussed in the appropriate module's Program Maintenance Manual where descriptions of module-specific supporting files and utilities are available. The Search and Print File Entries menu options are FileMan options that will not be maintained locally and are not discussed in this manual. The remainder of the options are discussed in this manual under the section number shown in parenthesis after the option name.

Appendix A cross references the option names with the option's text. Appendix B cross references the print templates with the ADMIN options that use them. Appendix C cross references the sort templates with the options. Appendix D cross references routine entry points with the options or routines that call them.

TABLE 3-1
ADMINISTRATION MODULE MENU OPTIONS

1 General System Tables

- 1 Set Up Organization Levels (4.2)
- 2 Set Up Site File (4.3)
- 3 Clinic Table Enter/Edit (4.4)
- 4 Edit Operation Class Name (4.5)
- 5 Edit Operation Subclass Name (4.6)
- 6 Set Up Operations File (4.7)
- 7 Set Up Occupation File (4.8)

2 Agency Functions

- 1 Create New Agency (5.2)
- 2 Agency Edit (5.3)
- 3 Inactivate Agency Unit (5.4)
- 4 Assign Agency Access to Users (5.5)
- 5 Assign Location for Agency Unit (5.6)
- 6 Inquiry for Agency Unit (5.7)
- 7 Agency Outline List (5.7)
- 8 Agency Units by Level (5.7)
- 9 Agency Units by Site (5.7)

3 Location Functions

- 1 Enter/Edit Location (6.2)
- 2 Assign Employee to Location (6.3)
- 3 Inactivate Location (6.4)
- 4 Assign Location for Agency Unit (5.6)

TABLE 3-1
ADMINISTRATION MODULE MENU OPTIONS
(CONTINUED)

4 Personnel Functions

- 1 Enter/Edit Employee (7.2)
- 2 Assign Employee to Location (6.3)
- 3 Terminate Employee (7.3)
- 4 Transfer Employee (Shop) (7.4)
- 5 Display Employee (7.5)
- 6 List Employees by Agency Unit (7.6)
- 7 List Employees by Location (7.7)
- 8 Enter/Edit Compensation Only Employee (7.8)
- 9 NCPDS Functions
  1. Load NCPDS Tape into Transaction File (7.9)
  2. Update Employee file from NCPDS Transactions (7.12)
  3. Edit NCPDS Transaction File (7.11)
  4. Delete NCPDS Transactions (7.13)
  5. Print Transactions (7.10)

5 Stressor Data Functions

- 1 Sample Units Enter/Edit (8.2)
- 2 Stressor File Enter/Edit (8.3)
- 3 Stressor Class Enter/Edit (8.4)
- 4 Clinical Data for Stressor Enter/Edit (8.5)

6 Module Tables and Files

- 1 Setup Environmental Tables (See the EE module Program Maintenance Manual)
  1. Setup PPE File
  2. Setup Respiratory Protection File
  3. Setup Collection Instrument Type
  4. Setup Frequency of Operations Eile
  5. Setup Laboratory File
- 2 Setup Medical Tables (See the MES module Program Maintenance Manual)
  1. Pre-exam Instructions Table Entry
  2. Medical Program Table Enter/Edit
  3. Medical Test Table Enter/Edit

TABLE 3-1
ADMINISTRATION MODULE MENU OPTIONS
(CONCLUDED)

3 ICC File Maintenance Functions (See the ICC Program Maintenance Manual)

1. OWCP Duty Station Agency File - Enter/Edit
2. Reporting Office for OWCP - Enter/Edit
3. Claim Type File - Enter/Edit
4. Nature of Injury File - Enter/Edit
5. Body Part File - Enter/Edit
6. Accident/Injury Type File - Enter/Edit
7. Agent of Accident File - Enter/Edit
8. Cause of Injury File - Enter/Edit
9. Source of Injury File - Enter/Edit
10. Adjudication Status File - Enter/Edit
11. OSH Injury Codes File - Enter/Edit
12. Pay Status File - Enter/Edit
13. Monthly Manhours Enter/Edit

4 Setup Deficiency Tables (See the HDA Program (Maintenance Manual)

1. Update Deficiency Type Table
2. Update Source of Deficiency Table
3. Inspector Table Update

7 Search and Print File Entries

1 Print File Entries
2 Search File Entries

4.0 GENERAL SYSTEM TABLES PROCESSES

4.1 Introduction

The Administration module maintains several files that are used as references by the rest of the OSHRKS modules. These files include the following:

- Organization Level
- Site
- Clinic Table
- Operation Class
- Operation Subclass
- Operation
- Occupation

The options that maintain these files are described in the following sections.

4.2 Set Up Organization Levels

4.2.1 Purpose

This option allows the user to create and update entries in the Organization Level file.

4.2.2 Overview

This edit option, illustrated in Figure 4-1, uses the routine T2JTLEV. It calls DIC for lookup, then calls T2GED and uses a GDR string to control editing. If the user enters a new organization level (i.e., a new level name), he/she will be prompted to enter a code for that level. Once a level has been entered, the user may edit only the level name. The user may print the contents of the Organization level file by entering routine T2JTLEV at tag PR.

4.2.3 Globals Referenced

The following global is read and updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Organization Level	---	1047	↑AGENCY(1047,	ADMIN	Update

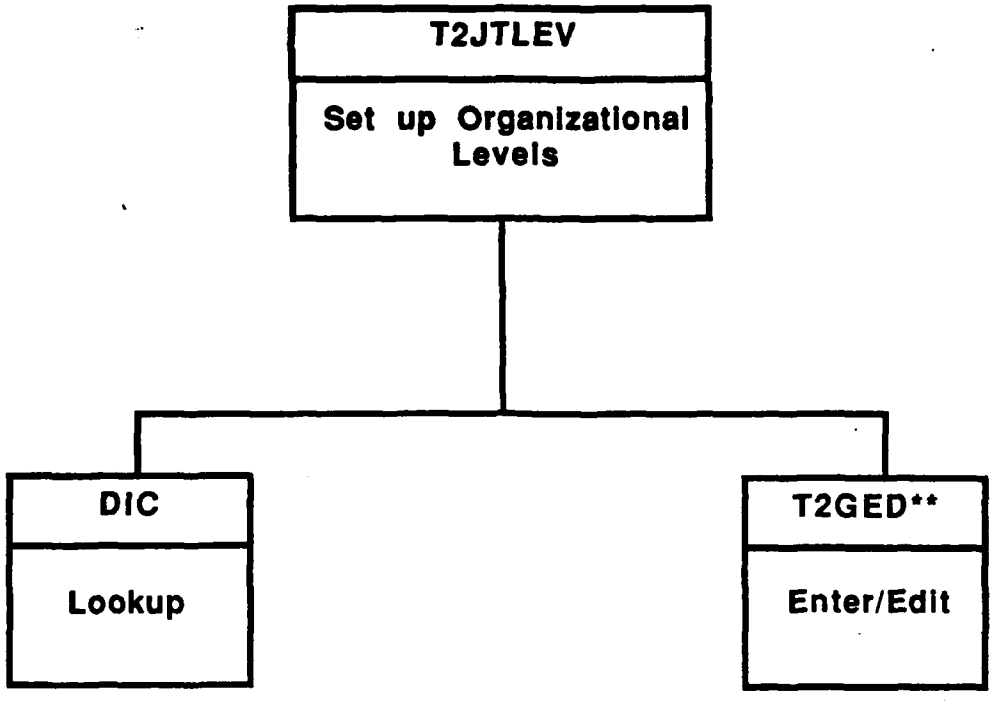

FIGURE 4-1
SET UP ORGANIZATION LEVELS OPTION
ROUTINE STRUCTURE

4.2.4 Variables

No variables used are unique to this option.

4.3 Set Up Site File

4.3.1 Purpose

This option creates and updates entries in the Site file.

4.3.2 Overview

This edit option, illustrated in Figure 4-2, calls routine T2GTSIT. This routine calls DIC for lookup, then calls T2GED which uses a GDR string to control editing. If the user adds a new site to file, he/she is prompted for an abbreviation for that site name. If the user edits a site, a check is made to see if the site has already been used as part of a location. If not, the user may edit the site abbreviation. If the site is being used in a location, the user may not edit the abbreviation. The user may print the contents of the Site file by calling routine T2GTSIT at line PR.

4.3.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Site	---	1041	↑AGENCY(1041,	ADMIN	Update

4.3.4 Variables

No variables used are unique to this option.

4.4 Clinic Table Enter/Edit

See Section 7.5 in the Medical Exam Scheduling Module Program Maintenance Manual.

4.5 Edit Operation Class Name

4.5.1 Purpose

This option creates and edits entries in the Operation Class file.

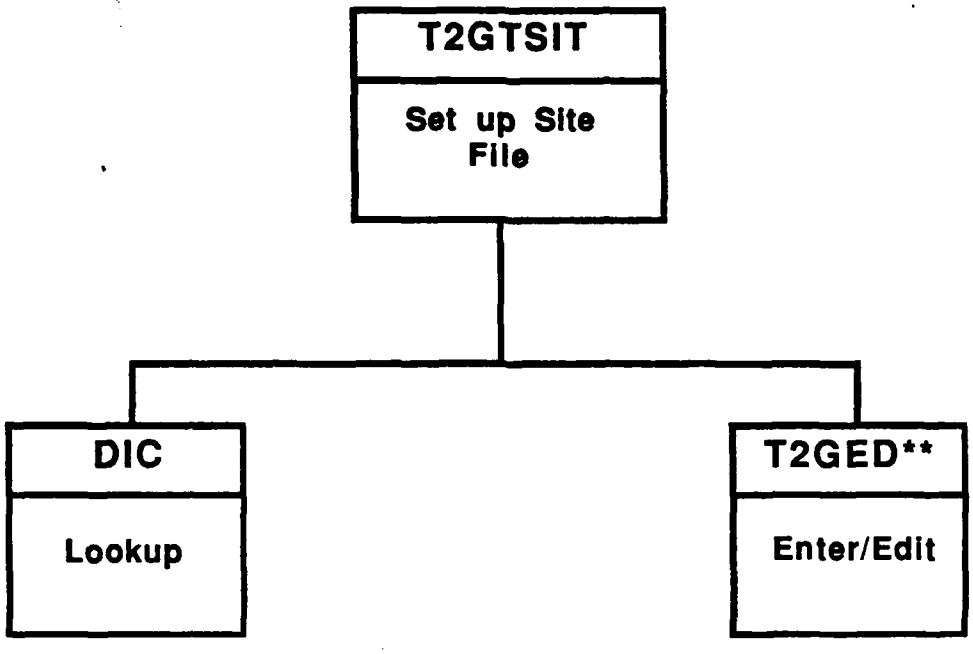

FIGURE 4-2
SET UP SITE FILE OPTION
ROUTINE STRUCTURE

4.5.2 Overview

This edit option calls routine T2GTOPR at line CLASS. This routine calls DIC and DIE, and uses a DR string to control editing. Figure 4-3 illustrates the routine structure.

4.5.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Operation Class	---	1152	↑DIZ(1152,	ADMIN	Update

4.5.4 Variables

The following variable is used in this option:

- GOPCL: The entry number in the Operation Class file for the current entry

4.6 Edit Operation Subclass Name

4.6.1 Purpose

This option creates and edits entries in the Operation Subclass file.

4.6.2 Overview

This edit option calls routine T2GTOPR at line tag OPSUB. This routine first does a lookup on the Operation Class file. It gets an existing entry or creates a new entry in that file. Then a lookup is done on the Operation Subclass file. Either an existing entry is selected or a new entry is created. The Class Name field is stuffed with the entry from the Operation Class file. T2GTOPR calls DIC and DIE, and controls editing with a DR string. The routine structure is shown in Figure 4-4.

4.6.3 Globals Referenced

The following globals are updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Operation Subclass	---	1153	↑DIZ(1153,	ADMIN	Update
Operation Class	---	1152	↑DIZ(1152,	ADMIN	Update

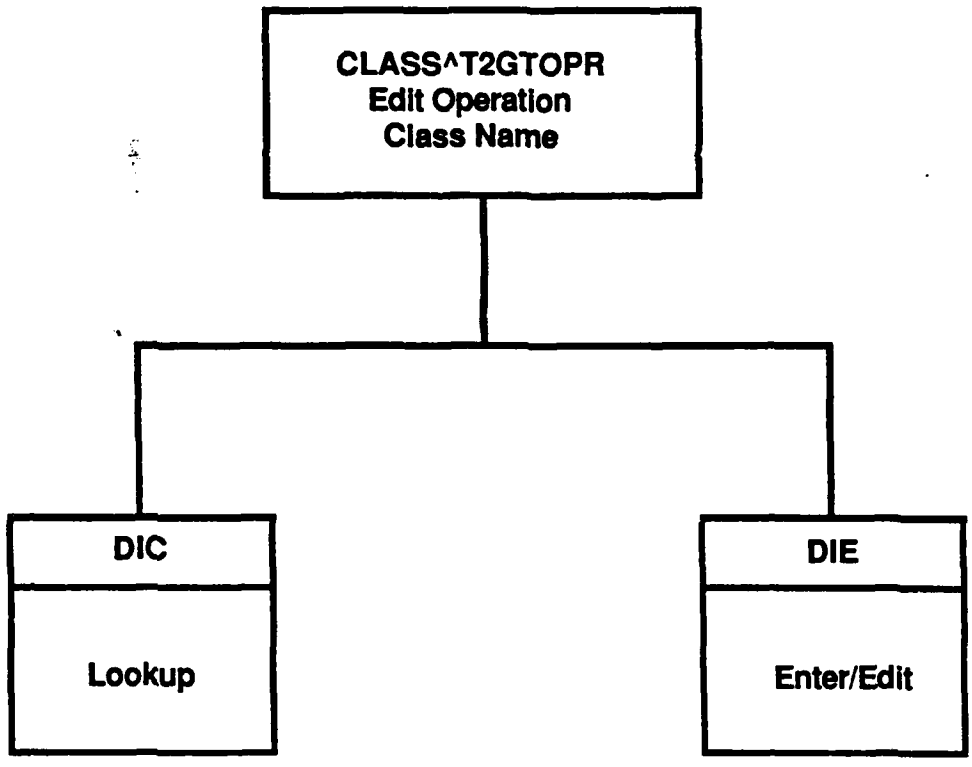

FIGURE 4-3
EDIT OPERATION CLASS NAME OPTION
ROUTINE STRUCTURE

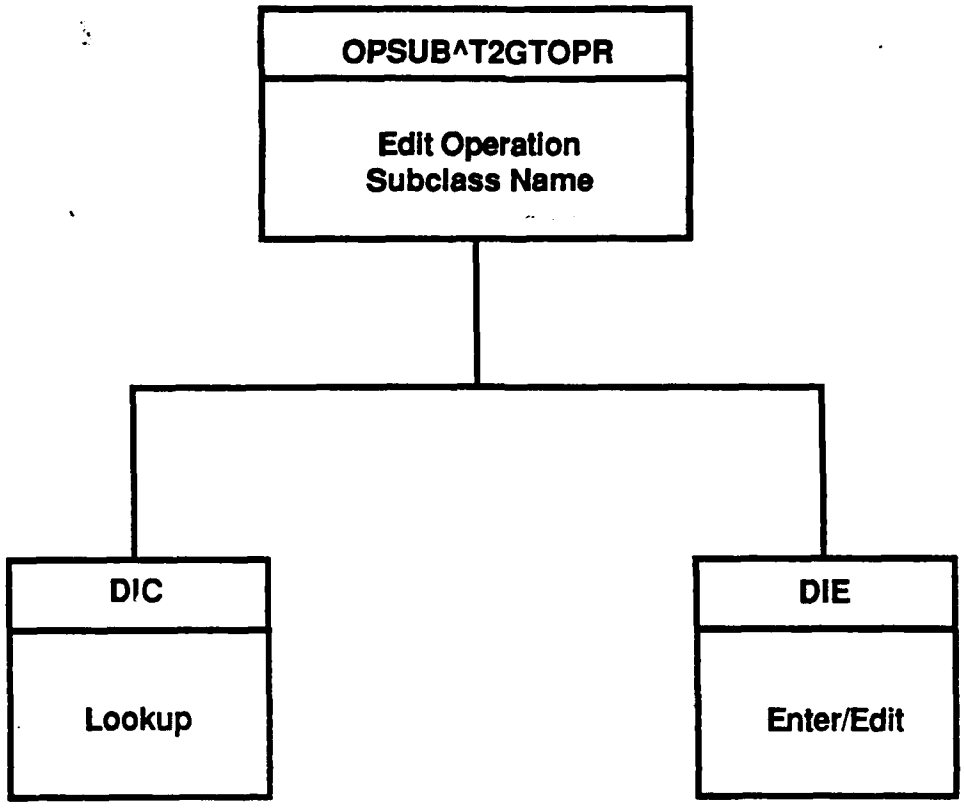

FIGURE 4-4
EDIT OPERATION SUBCLASS NAME OPTION
ROUTINE STRUCTURE

4.6.4 Variables

The following variable is used in this option:

- GOPCL: The entry number in the Operation Class file for the current entry

4.7 Set Up Operation File

4.7.1 Purpose

This option creates and edits entries in the Operation File.

4.7.2 Overview

This edit option, illustrated in Figure 4-5, calls routine T2GTOPR at line ENT. This routine calls DIC for lookup and T2GED for editing. The user may print the contents of the Operation file by calling routine T2GTOPR at line PR.

4.7.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Operation	---	1087	↑DIZ(1087,	ADMIN	Update

4.7.4 Variables

The following variable is used in this option:

- GOPCL: The entry number in the Operation Class file for the current entry

4.8 Set Up Occupation File

4.8.1 Purpose

This option creates and edits entries in the Occupation file.

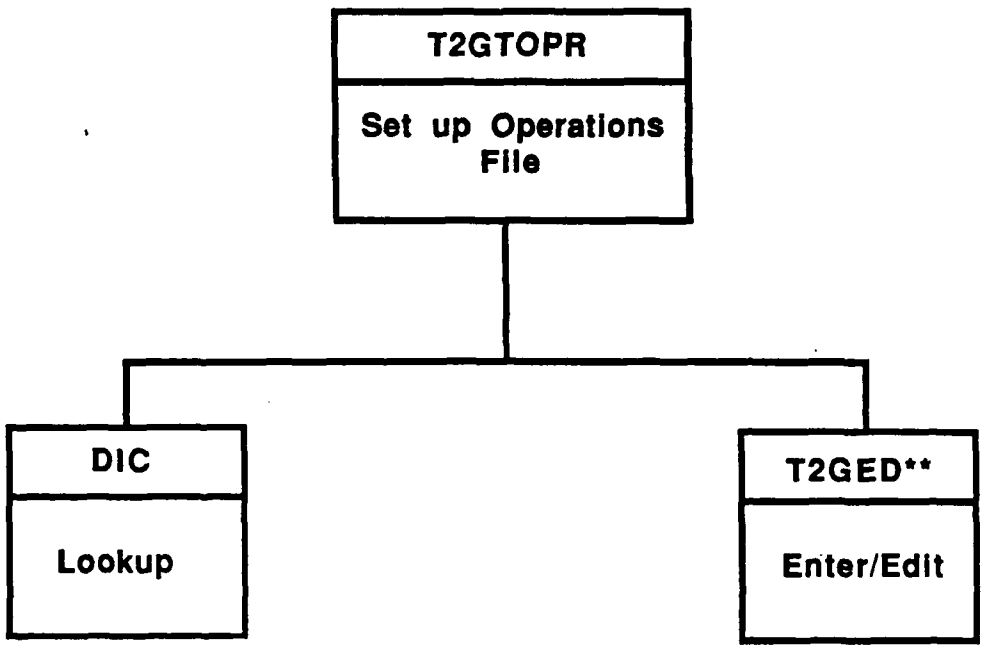

FIGURE 4-5
SET UP OPERATIONS FILE OPTION
ROUTINE STRUCTURE

4.8.2 Overview

This edit option, illustrated in Figure 4-6, calls routine T2GTOCC at line ENT. This routine first does a lookup to determine whether the occupation type "terminated" is in the file. If not, the entry is created. Then T2GTOCC calls DIC to create a new entry or select an existing entry. Following this, T2GED is called to edit the current entry. The user may print the contents of the Occupation file by an option that calls T2GTOCC at line PR.

4.8.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Occupation	---	1001	↑DIZ(1001,	ADMIN	Update

4.8.4 Variables

There are no variables used that are unique to this option.

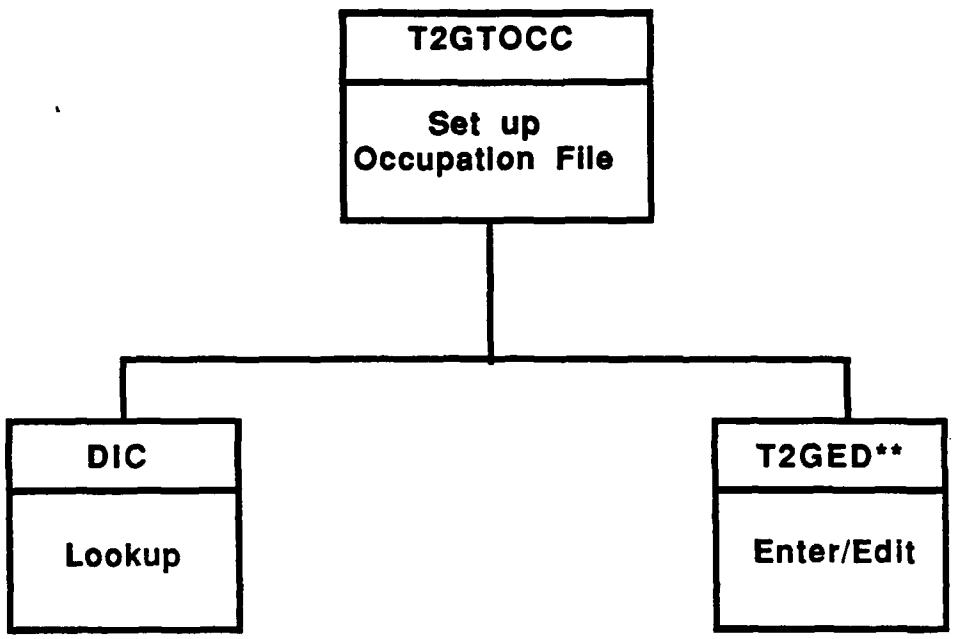

FIGURE 4-6
SET UP OCCUPATION FILE OPTION
ROUTINE STRUCTURE

5.0 AGENCY PROCESSES

5.1 Introduction

The organization structure of each activity using OSHRKS must be entered into the Agency Unit file. In OSHRKS, the terms "agency" and "activity" are used interchangeably. The term "agency unit" refers to a specific subpart of the agency, usually a shop, department, or group. The Agency Unit file contains data about each agency and each agency unit within the agency. One of the more important items of data that is recorded is the reporting hierarchy. Figure 5-1 illustrates an example of a simple agency structure. The following sections discuss the options that control the Agency Unit file.

5.2 Create New Agency

5.2.1 Purpose

The Create New Agency function is used to set up the top-most level of an agency in the Agency Unit file and, once this is done, to enter the subordinate levels of the agency. There are variables in the top-most level entry that are unique to that position in the hierarchy, hence this top-most level entry requires a special input sequence.

5.2.2 Overview

The routine T2JENL is invoked to perform the Create Agency option. Figure 5-2 illustrates the routine structure. The first step in this routine is to set the user's agency access string to a comma (",") if it does not already exist. Then the routine determines if the Site file, Organization Level file, and Clinic file exist. If these files do not exist, the user is given a message that these files must be created before an agency can be entered. If the routine continues, DIC is called to look up the agency. The screen on the lookup limits the entries found to those that are at the top level (i.e., the numeric portion of Level Code equals 1) and are null in the Commander/Office/Department field. If no entry is found or created, the routine exits. The agency access string is updated with access authorization for the new agency if it is not already in the string. (This means that the security for the option must limit users to those who are truly authorized and trained to use this option.) If the agency selected is not a newly created one, the routine goes to the Agency Edit routine (EDIT↑T2JEN). The agency entry is locked with a timeout; the entry is killed with DIK if it cannot be locked. T2GED is called for entry of the agency fields. Then the Counters file is locked, used to create the new Astring field for this agency, and unlocked. The Astring and agency are filed using DIE and a history entry is filed using T2JEN1 at line HF. DIC and DIE are used to create the required agency units:

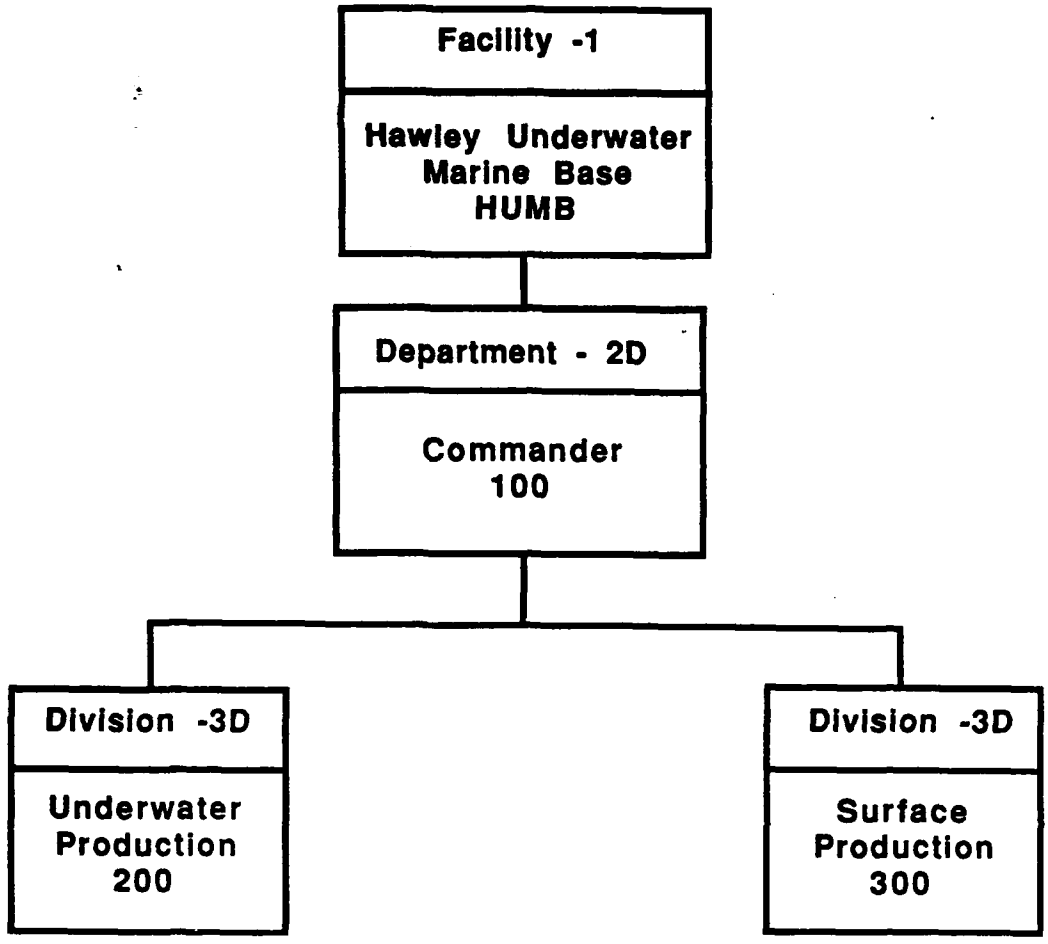

FIGURE 5-1
SAMPLE AGENCY STRUCTURE

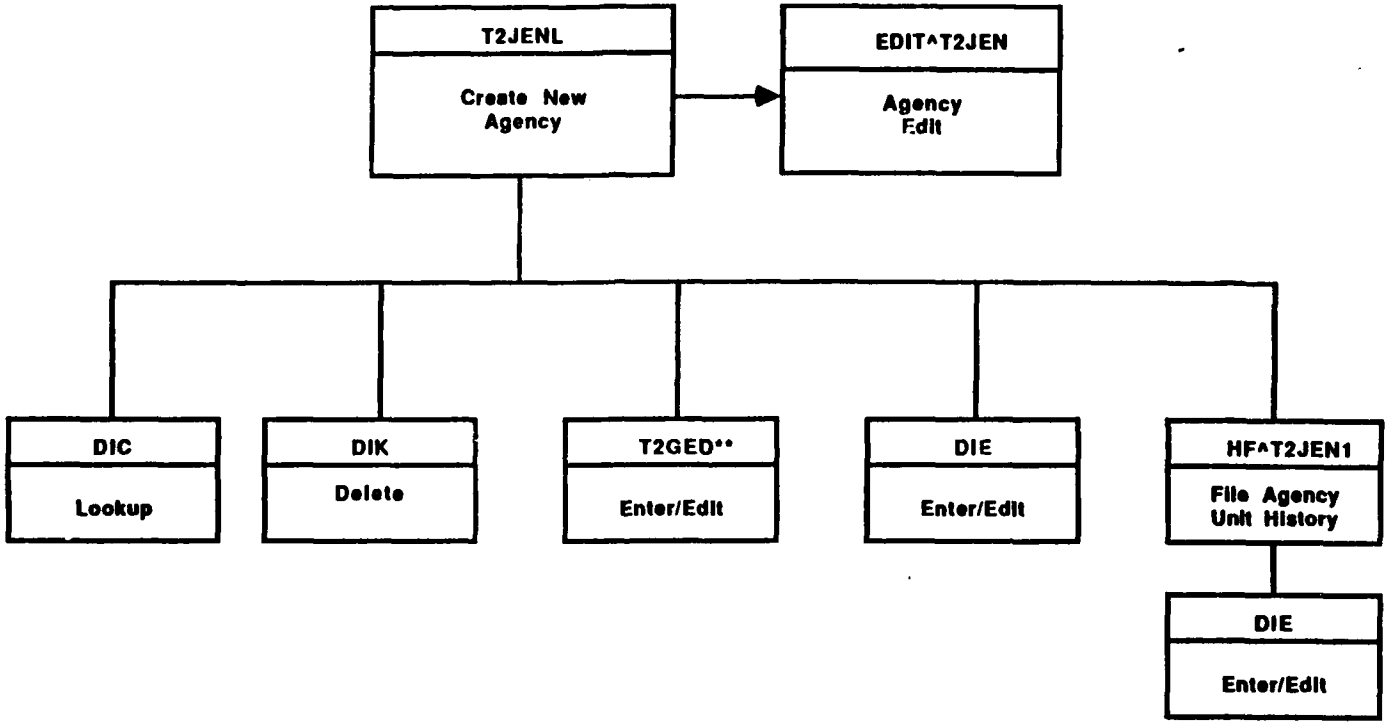

FIGURE 5-2
CREATE NEW AGENCY OPTION
ROUTINE STRUCTURE

Prehire, Termination, and Compensation Only, for this agency. The routine then continues processing by going to EDIT↑T2JEN for the entry of additional agency units. For a detailed explanation of the processing from this point, see Section 5.3.

5.2.3 Globals Referenced

The following globals are read or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Update
Counters	---		↑AGENCY(0,"COUNT",	ADMIN	Update
Organization Level	---	1047	↑AGENCY(1047,	ADMIN	Read
Site	---	1041	↑AGENCY(1041,	ADMIN	Read
Clinic	---	1138	↑MED(0,	MES	Read

5.2.4 Variables

The following variables are used in T2JENL:

- DUZ("AG"): The user access string containing the pointer values of each agency the user may access
- J1: First character of the Astring field
- J2: Second character of the Astring field
- JDA: Pointer value (DA) of the agency
- JLB: Set to 1
- JLE: Set to 1
- JLL: Astring value for the agency
- JNM: Code/Abbreviation of the agency
- JSTR: Astring value of the agency
- JTL: Variable used to set up each of the three agency units: Prehire, Termination, Compensation Only, that are required for each agency

5.3 Agency Edit

5.3.1 Purpose

The Agency Edit option is used to modify the contents of the agency description in the Agency Unit file. This modification includes the ability to change the value of specific fields for an agency unit or to redefine the hierarchy of an agency by adding and deleting subordinate units from the agency tree.

5.3.2 Overview

The routine T2JEN is called for the Agency Edit option. Figure 5-3 shows the routine structure. The entry point EDIT in this routine is used from the Create Agency option to enable the user to define subordinate units in the agency or edit previously entered data.

T2JEN determines if the Site file, Organization Level file, and Clinic file exist. If these files do not exist, the user is given a message that these files must be created before an agency can be entered. If the routine continues, DIC is called to look up the top level of the agency. The DIC("S") screen on the lookup limits the entries to those that are at the top level (the numeric portion of Level Code equals 1), and are null in the Commander/Office/Department field. If no entry is found, the routine exits. The variables specific to the agency are set before the tag EDIT is encountered. Beginning at the EDIT tag, the routine locks the agency file entry for the agency being worked on. If the lock command fails the timeout constraint, a message is displayed to the user, and the routine exits. At the tag ASK, the user is given two options for entry of data: "add/edit AGENCY UNITS in this AGENCY" and "edit this AGENCY". If the user does not choose either of these activities, the routine exits. If the user wishes to "add/edit an AGENCY UNIT", the routine T2JEN1 is called at the tag ENT. If the user wishes to "edit this AGENCY", variables are set, and DIQ is invoked to display the agency data to the user. (This action applies only to the top level agency unit data.) Then T2GED is called to perform the enter/edit for the appropriate fields of the agency entry, and T2JEN1 is called at the tag HF to file historical data. If the user has changed the UIC field for the entry, DICD at the tag WAIT is called to notify the user that there will be a pause while the system processes the file changes. Then the routine steps through the Employee file using the "AL" cross-index (Agency pointer). Old "AN" cross-index values are deleted and new "AN" cross-index values are created, for all employees who have a Social Security Number (SSN) field, and who are not filed as "Comp Only". (The "AN" cross-reference of the Employee file is a combination of agency UIC and employee SSN which is used by the Update Employee File from NCPDS Transactions option to identify an employee.)

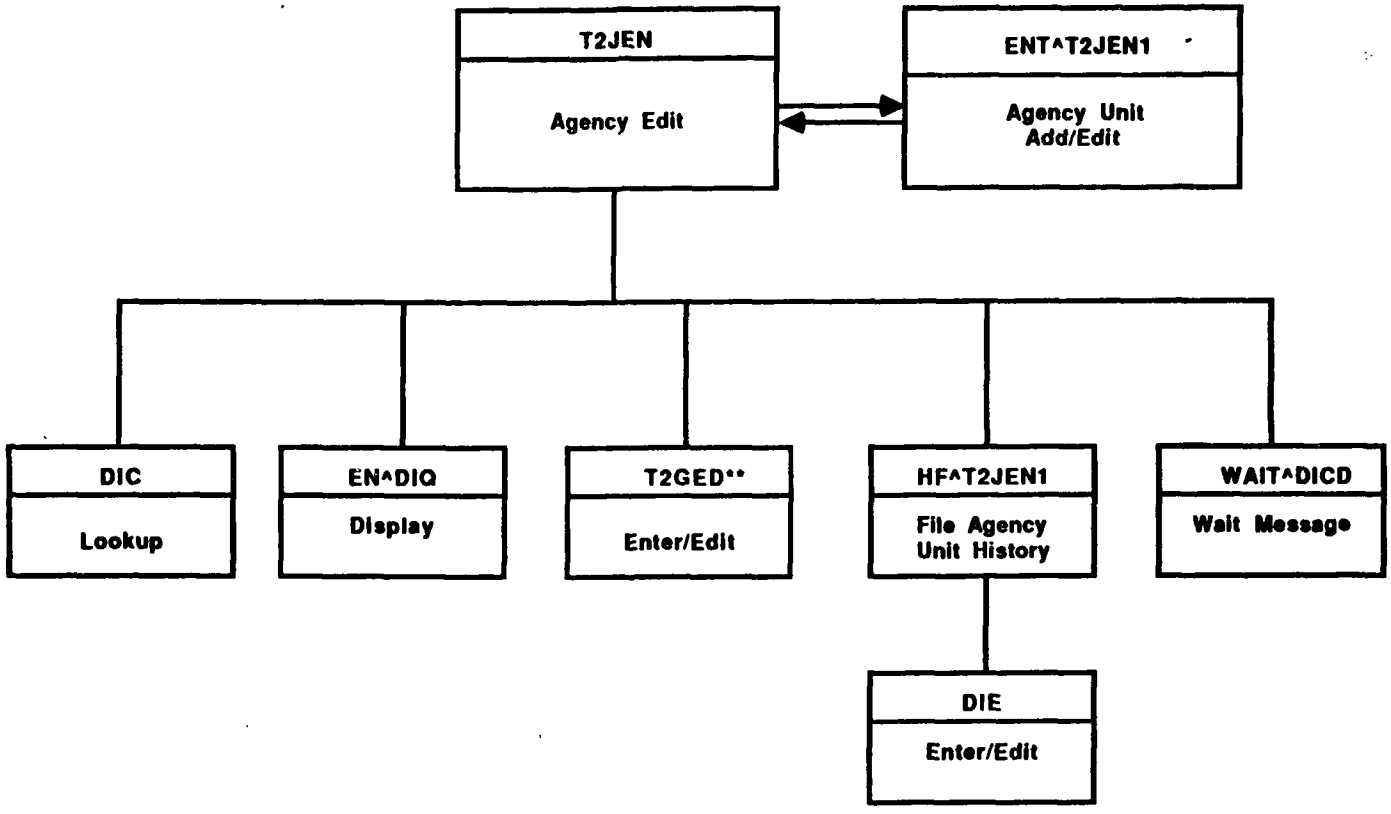

FIGURE 5-3
AGENCY EDIT OPTION
ROUTINE STRUCTURE

T2JEN1 sets a screen (DIC("S")) to limit all lookups within the session to agency units that are contained in the selected agency and are currently active (contain no Inactive Date field). The prompt for the user is set to special text (DIC("A")) asking the user to select the agency unit for which subordinate units are to be defined. This unit is considered to be the ascendant unit for agency units that are immediately subordinate to it. DIC is used to perform the lookup for the ascendant unit. This agency unit must already be on file, since the agency organization must be defined in a top-down fashion. If no agency unit is selected, the routine exits. Variables which are used to define features of subordinate units are then set. Starting at line tag NX, the routine begins the process of capturing the data for each subordinate unit within the selected ascendant unit's organizational structure. For each subordinate unit defined, the DIC("S") screen is set on lookup to screen for entries that are in the correct agency, are for the selected ascendant unit, and are not top level agency units. DIC performs the lookup with LAYGO. If the lookup returns no entry, the routine does tag LS to set the last used pair of letters for the Astring calculation into the ascendant unit's entry, and the routine loops to the prompt for selecting a new ascendant unit. If the selected agency unit is already on file, the routine goes to the tag EDIT. For new agency units, the Astring is calculated, and T2GED is called to control the entry of data. If the data entry sequence has been exited prematurely, the routine exits. Otherwise, the JLAST variable is set to the last Astring letter pair used. HF is done to file the historical data, and the routine loops back to the tag NX. In the paragraph HF, the routine checks the Effective Date field, if there is not Effective Date on file, the history is not created. Then the routine sets up the variables for the use of *DIE*. If the Agency History entry is a new one, the .01 value is added to the DR string. DIE is used to file the Agency History subfile entry.

5.3.3 Globals Referenced

The following globals are read or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Update
Organization Level	---	1047	↑AGENCY(1047,	ADMIN	Read
Site	---	1041	↑AGENCY(1041,	ADMIN	Read
Clinic	---	1138	↑MED(0,	MES	Read

5.3.4 Variables

The following variables are used in this option:

- DUZ("AG"): The user access string containing the pointer values of each agency the user may access
- J1: First character of the Astring field
- J2: Second character of the Astring field
- JASC: Pointer value of ascendant agency unit
- JAST: Astring value for ascendant agency unit
- JDA: Pointer value (DA) of the agency
- JLAST: Last letter pair used for subordinate units of the ascendant agency unit
- JLB: Numeric value of Level Code for ascendant unit plus 1 (minimum numeric value allowed for use as Level Code for subordinate units)
- JLE: Numeric value of Level Code field for agency unit being edited
- JLL: Astring value for the agency
- JNM: Code/Abbreviation of the agency
- JSTR: Astring value of the agency
- JSX: ,0) node of newly created or edited agency unit
- JU: New value of the top level agency UIC field
- JUIC: Original value of the top level agency UIC field

5.3.5 Remarks

The Agency History subfile uses DINUM with an inverted date to file each entry in reverse chronological order and to ensure that each entry is for a single unique date.

5.4 Inactivate Agency Unit

5.4.1 Purpose

This option inactivates an agency unit to prevent employees and locations from being assigned to that agency unit.

5.4.2 Overview

This edit option, illustrated in Figure 5-4, calls routine T2JI at the tag ENT. The user is prompted for the name of the agency unit he/she wants to inactivate using DIC. The routine finds all the agency units whose Astring starts with the same letters as the Astring of the selected agency. The entry number for each of these agency units is loaded into the scratch table JSB. Next, the routine searches the employee file for employees assigned to any of the agency units in the JSB table. If any such employees are found, a warning message is displayed and control passes out of the routine. If no employees are assigned to any of the agency units, the routine searches the Agency Unit file to determine if any of the agency units has locations assigned to it. If so, a warning message is displayed and control passes out of the routine. If no locations are assigned to any of the agency units, the routine lists all agency units that will be inactivated and prompts the user to proceed. The user may exit the routine at this point. Otherwise the routine sets the Inactivate Date field of the Agency Unit file to the current date for each agency unit in the JSB table, using T2GED and DIE.

5.4.3 Globals Referenced

The following globals are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Update
Employee	---	1004	↑EMPLOY(	ADMIN	Read

5.4.4 Variables

The following variables are used in this option:

- JAST: The Astring of the original agency unit to be inactivated
- JC: A flag used to warn the user that an employee or location is assigned to the agency unit to be inactivated

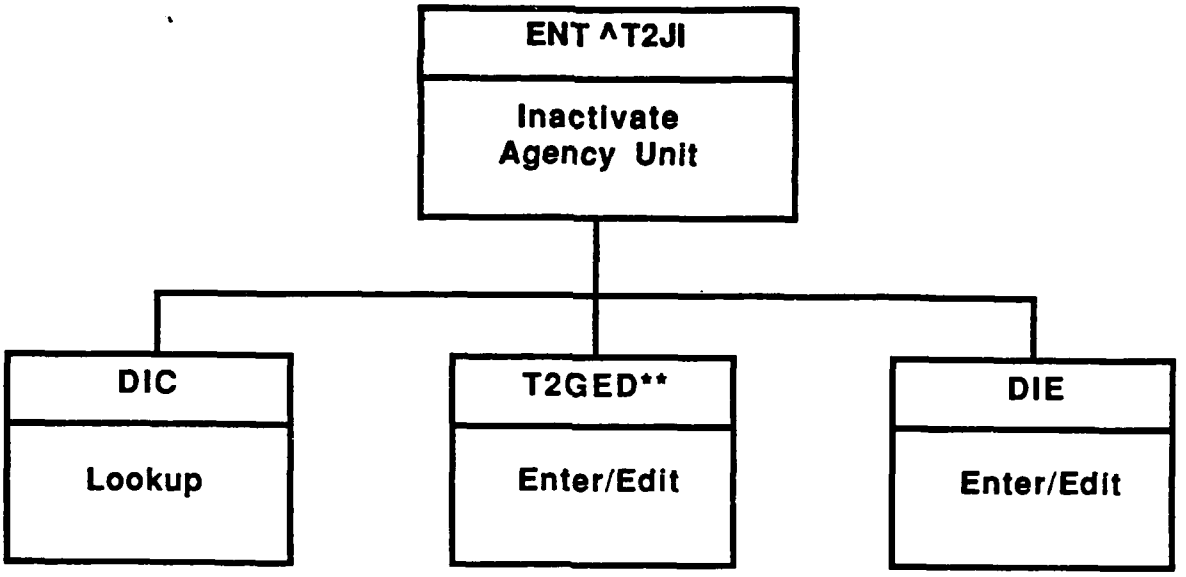

FIGURE 5-4
INACTIVATE AGENCY UNIT OPTION
ROUTINE STRUCTURE

- JLN: The length of the Astring of the original agency unit to be inactivated
- JNX: The Astring of the agency unit to be inactivated; this variable changes as the routine searches the Agency Unit file index
- JSB: An array whose subscripts are the entry numbers of the agency units to be inactivated
- JSUB: The entry number of each agency unit to be inactivated

5.5 Assign Agency Access to Users

5.5.1 Purpose

This option updates the User file with the agencies to which a user is allowed access.

5.5.2 Overview

This edit option calls routine T2GUSEN which calls DIC and DIE. Editing is controlled with a DR string. Figure 5-5 shows the routine structure.

5.5.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
User	---	3	↑DIC(3,	ADMIN	Update

5.5.4 Variables

There are no variables used that are unique to this option.

5.6 Assign Location for Agency Unit

5.6.1 Purpose

This option updates the Agency Unit file with the locations in which the agency unit(s) work. This data is used for documentation purposes only and does not affect processing of personnel or other types of data in the system.

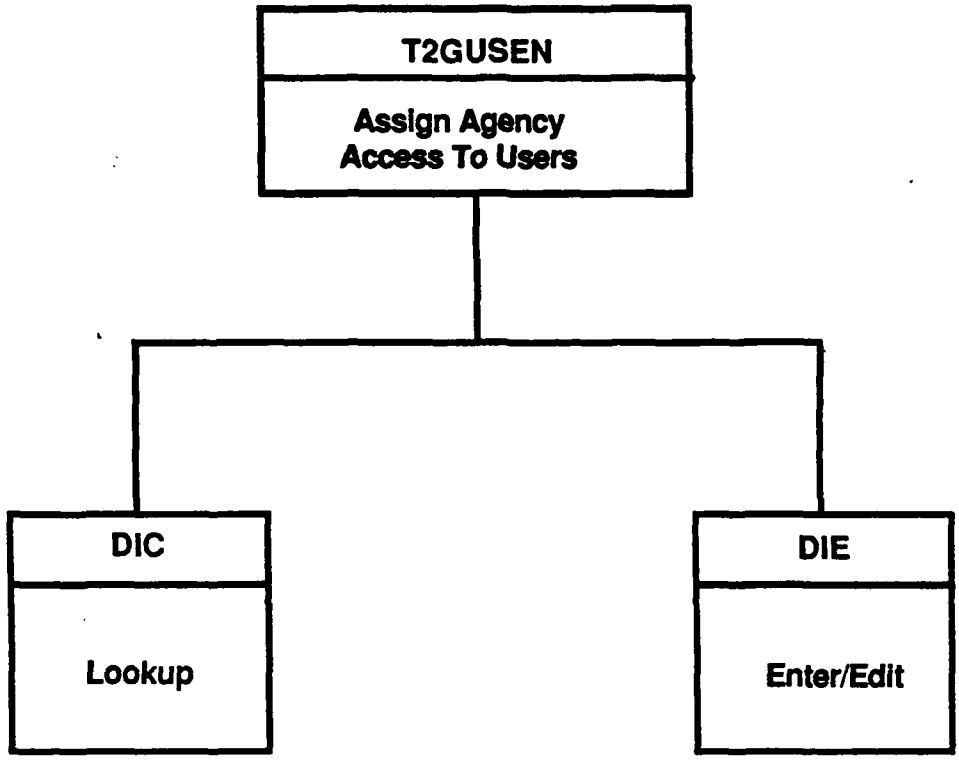

FIGURE 5-5
ASSIGN AGENCY ACCESS TO USERS OPTION
ROUTINE STRUCTURE

5.6.2 Overview

This edit option, illustrated in Figure 5-6, calls routine T2JA at tag ENT. The user is asked whether he/she wishes to enter all locations for a single agency or all agencies for a single location. Routine T2GL is called to get a location (see Section 9.6) and DIC is called to get an agency. Then the routine T2JA passes control to tag DET. If the location has been entered previously for that agency unit, the user is given the opportunity to remove the location from the agency unit at tag EDIT. If he/she removes the location, the routine T2JLH is called to update the Location History subfile in the Agency Unit file and routine DIE deletes the Current Location subfile entry. If the location does not exist for the selected agency unit, T2GED enters the location in the Current Location subfile and T2JLH updates the Location History subfile. After the location has been entered or removed from the agency unit, the user is prompted for another location and agency unit.

Routine T2JLH is entered at the tag ENT. If there is no entry in the Location History subfile of the Agency Unit file for the selected location, the location is added. The effective date of the transaction and the termination status are also added.

5.6.3 Globals Referenced

The following globals are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Agency Unit	Location	1074.01	↑AGENCY(0,#,1,	ADMIN	Update
Agency Unit	Location History	1074.02	↑AGENCY(0,#,2,	ADMIN	Update

5.6.4 Variables

The following variables are used in this option:

- LN: The entry number of the location in the Location subfile of the Agency Unit file
- PL: The entry number and first piece of the zero node for the selected location in the Location file
- PN: The entry number of the selected agency in the Agency Unit file

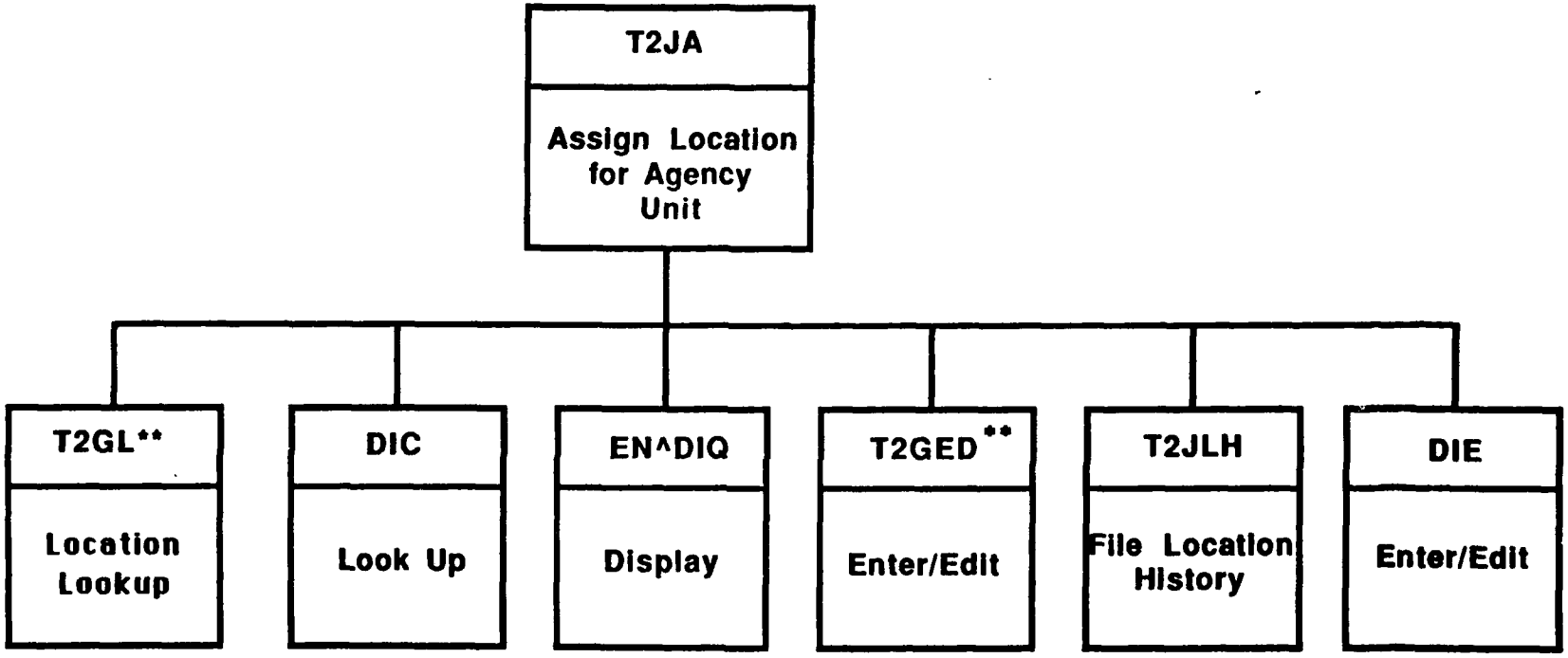

FIGURE 5-6
ASSIGN LOCATION FOR AGENCY UNIT OPTION
ROUTINE STRUCTURE

- PS(0): The effective date of the location assignment

- PS(1): The termination status of the location; set to null for new assignment, "T" for terminated or removed assignment

5.7 Agency Output Options

5.7.1 Purpose

The following options list agency unit data using a variety of formats and selection criteria:

Inquiry for Agency Unit

Agency Outline List

Agency Units by Level

Agency Units by Site

5.7.2 Overview

Routine T2JR1 performs the Inquiry for Agency Unit option. Routine DIC is called to select an agency unit. Then the user is asked whether descendant (subordinate) units are to be included and the report is produced using routine DIP and print template AGENCY UNIT. Figure 5-7 shows the routine structure.

The Agency Outline List is produced by routine T2JR2, which calls routine T2JLMF to select an agency unit and prints an indented list of the agency unit selected and any subordinate units. The routine structure is presented in Figure 5-8. As used here, routine T2JLMF is a simple lookup using routine DIC.

Routine T2JRL performs the Agency Units by Level option. Routine DIC is used to select an agency. DIC is used at line tag IX to select an organization level. The user is informed when there are no agency units in the file for the selected level. Otherwise, routine DIP is called to produce the list using print template AGENCY UNIT and sort template AGENCY LEVEL. Figure 5-9 illustrates the routine structure.

The Agency Units by Site option is performed by routine T2JRS, which calls routine DIC to select an agency and a site and routine DIP to produce the list using print template AGENCY UNIT and sort template AGENCY SITE. The routine structure is shown in Figure 5-10.

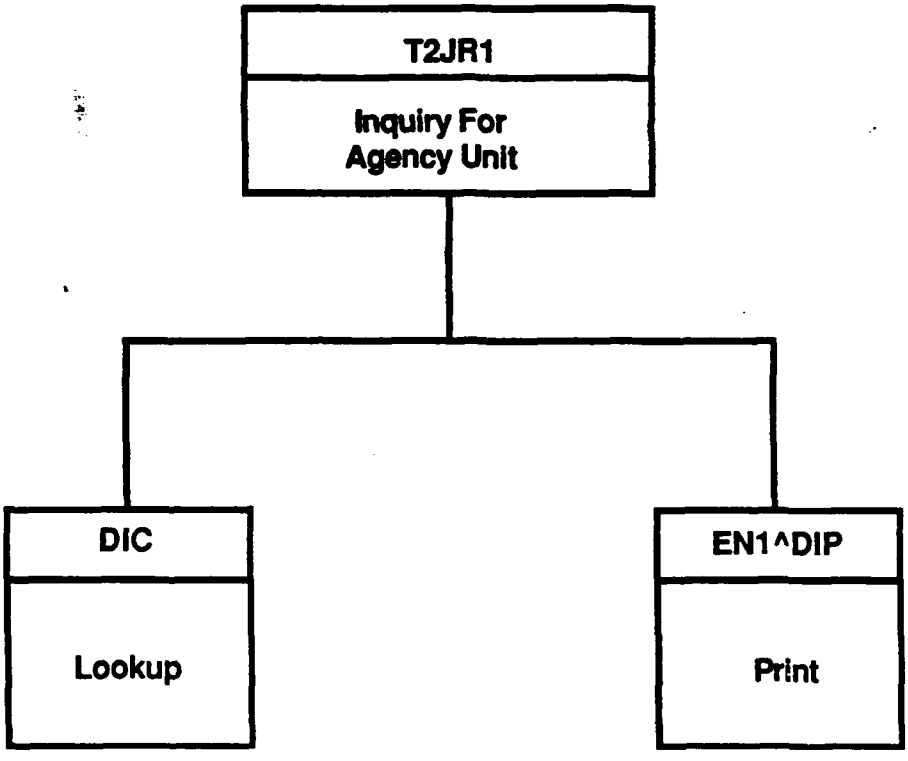

FIGURE 5-7
INQUIRY FOR AGENCY UNIT OPTION
ROUTINE STRUCTURE

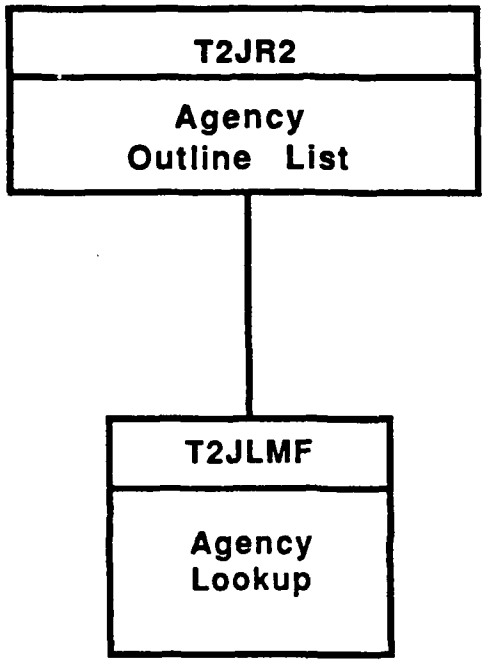

FIGURE 5-8
AGENCY OUTLINE LIST OPTION
ROUTINE STRUCTURE

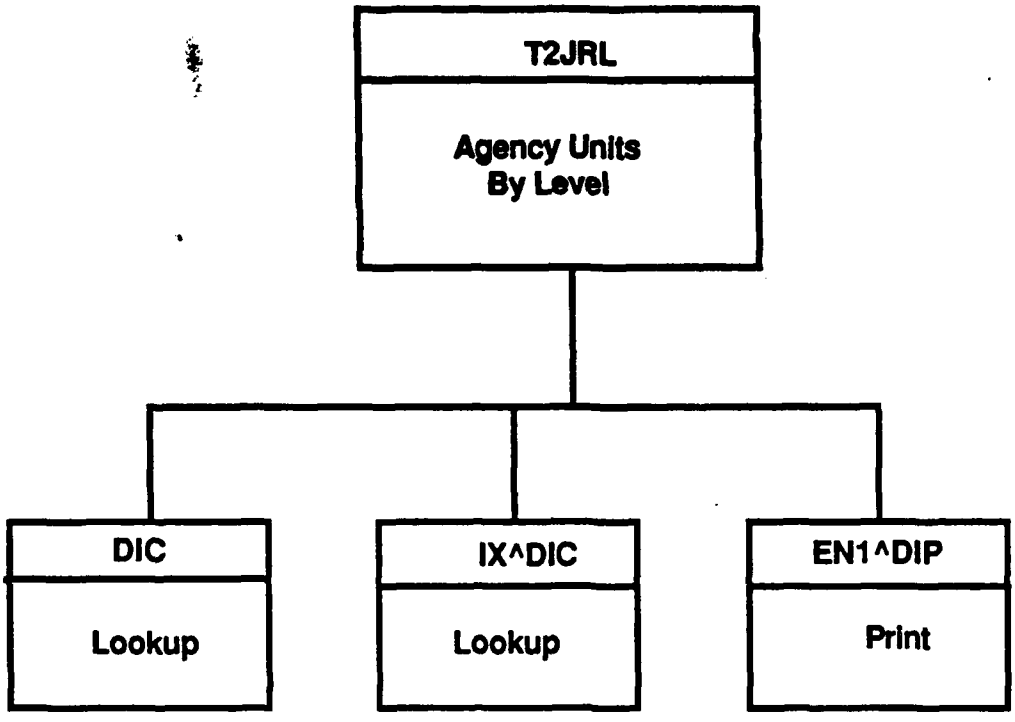

FIGURE 5-9
AGENCY UNITS BY LEVEL OPTION
ROUTINE STRUCTURE

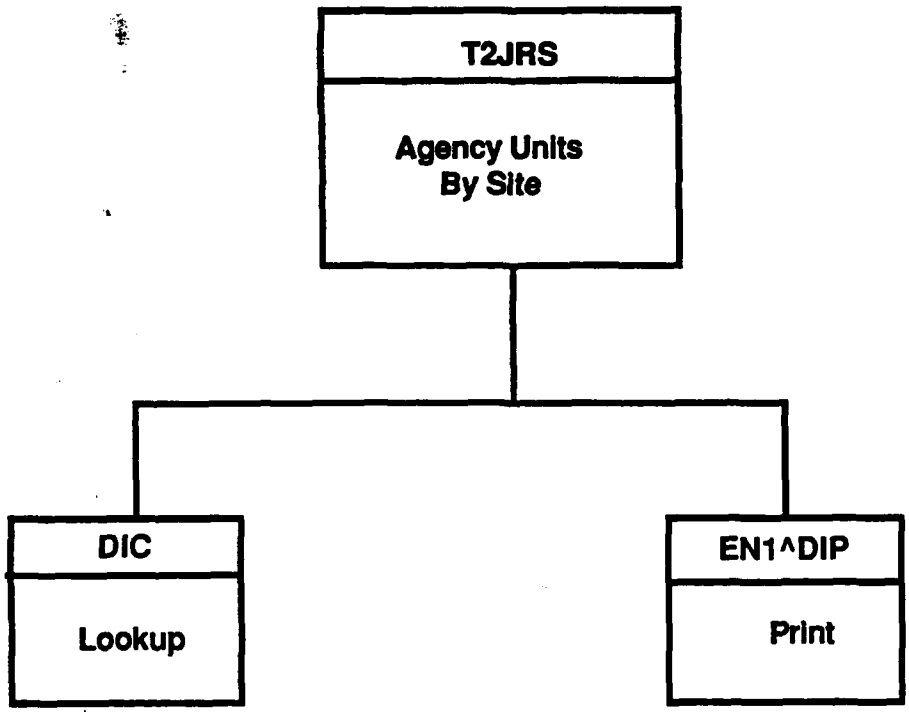

FIGURE 5-10
AGENCY UNITS BY SITE OPTION
ROUTINE STRUCTURE

5.7.3 Globals Referenced

The following globals are read by this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Organization Level	---	1047	↑AGENCY(1047,	ADMIN	Read
Site	---	1041	↑AGENCY(1041,	ADMIN	Read

5.7.4 Variables

In addition to standard FileMan variables, routine T2JR1 uses the following variables:

- JAGN: The Code/Abbreviation and Name of an Agency Unit file entry
- JSTR: The Astring field for an Agency Unit file entry
- JWITH: A value of one indicates that subordinate units are to be listed; a value of zero indicates they are not to be listed
- JY0: The zero node for an Agency Unit file entry

Routine T2JR2 uses the following variables:

- JAGU: The Astring field for the selected Agency Unit file entry
- JL: The length of the Astring in variable JAGU
- X: The Astring field for an Agency Unit file entry
- Y: An Agency Unit file entry number
- Y(0): The zero node of an Agency Unit file entry

None of the variables in T2JLMF affect the processing except for the standard FileMan variables.

The following variables are used by routines T2JRL and T2JRS in addition to standard FileMan variables:

- JAGN: An agency name
- JDA: An Agency Unit file entry number for an agency
- JY0: The zero node of an agency entry

6.0 LOCATION PROCESSES

6.1 Introduction

Location data is stored in the Location file. Locations in OSHRKS are composed of four pieces of data, separated by commas: (1) site abbreviation, (2) location, (3) sublocation, and (4) area. A site is the general geographic place where a facility is located, a survey was conducted, or an injury incurred, e.g., Mare Island, Bethesda Naval Hospital, Point Loma.

6.2 Enter/Edit Location

6.2.1 Purpose

This option creates entries in the Location file and edits the name and effective date of each entry.

6.2.2 Overview

This option is performed by routine T2GLE. Figure 6-1 shows the routine structure. First, routine T2GL is called to select or add a Location file entry (see Section 9.6). If an existing entry is selected, the user is asked whether the description is to be edited. If so, routine DIE is called to edit the name and effective date. A historical entry is created using routines DIC and DIE to file a Location History subfile entry for the effective date.

6.2.3 Globals Referenced

The following globals are updated/read in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Location	---	1073	↑AGENCY(1073,	ADMIN	Update
Site	---	1041	↑AGENCY(1041,	ADMIN	Read

6.2.4 Variables

The following variables are used in this option:

- GDT: Effective date of location entry
- GLAY: Flag that indicates LAYGO is allowed on a lookup
- GSI: First piece of the location name (i.e., site)

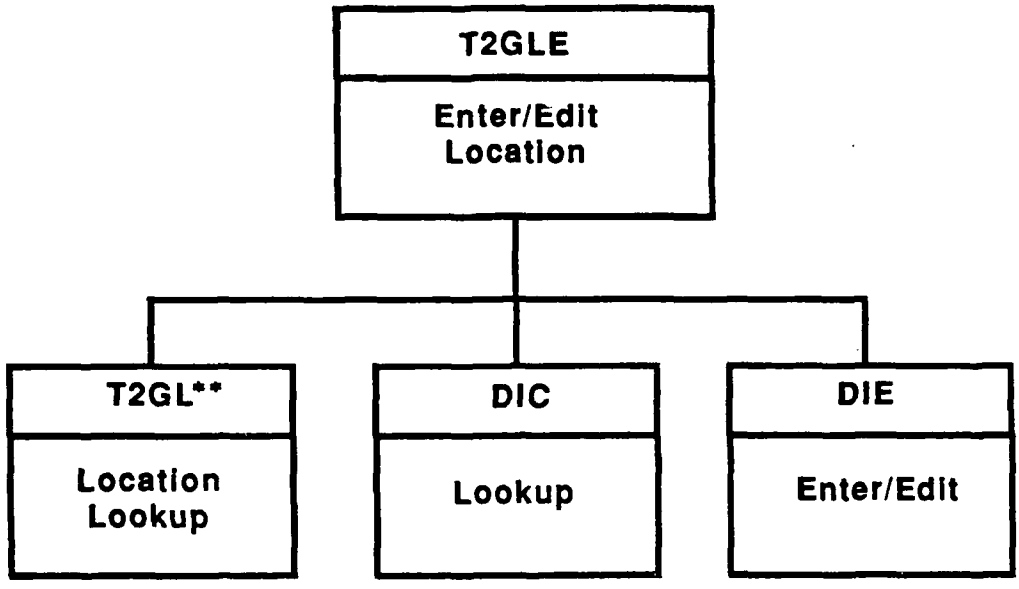

FIGURE 6-1
ENTER/EDIT LOCATION OPTION
ROUTINE STRUCTURE

6.3 Assign Employee to Location

6.3.1 Purpose

This option updates the Employee file with the locations to which an employee is assigned.

6.3.2 Overview

This edit option, illustrated in Figure 6-2, calls routine T2PA at tag ENT. The user is asked whether he/she wishes to enter all employees at a given location or all locations for a given employee. After the user has entered the location and employee using T2GL and T2PL, respectively, (Sections 9.6 and 9.3), the routine passes control to DET. If the location has been entered previously for that person, the user is given the opportunity to remove the location from the person at the tag EDIT. If he/she removes the location, routine DIE is used to enter an effective date and delete the Current Location subfile entry and routine T2PH is called at line tag LOC to update the Location History subfile in the Employee file. When an existing location is not removed, the user is given the opportunity to edit the Hours per Week. Routine DIE does the prompting for the effective date and hours per week. If a new location is added for the employee, using DIE, T2PH is also called to update the Location History subfile. After the location has been entered or removed from the Employee file, the user is prompted for another location or employee, depending on what the user specified at the beginning of T2PA.

Routine T2PH is a utility routine that updates several history subfiles in the Employee file. When the routine is called at the tag LOC, the routine updates the Location History subfile. First it forces a new entry with a call to DIE using a DR string containing the .01 field. Then it updates the entry with an effective date, hours per week, and termination status.

6.3.3 Globals Referenced

The globals referenced in this option are:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Employee	Current Location	1004.03	↑EMPLOY(#,4,	ADMIN	Update
Employee	Location History	1004.04	↑EMPLOY(#,5,	ADMIN	Update
Location	---	1073	↑AGENCY(1073,	ADMIN	Read
Medical Program	---	1088	↑MED(1088,	MES	Read

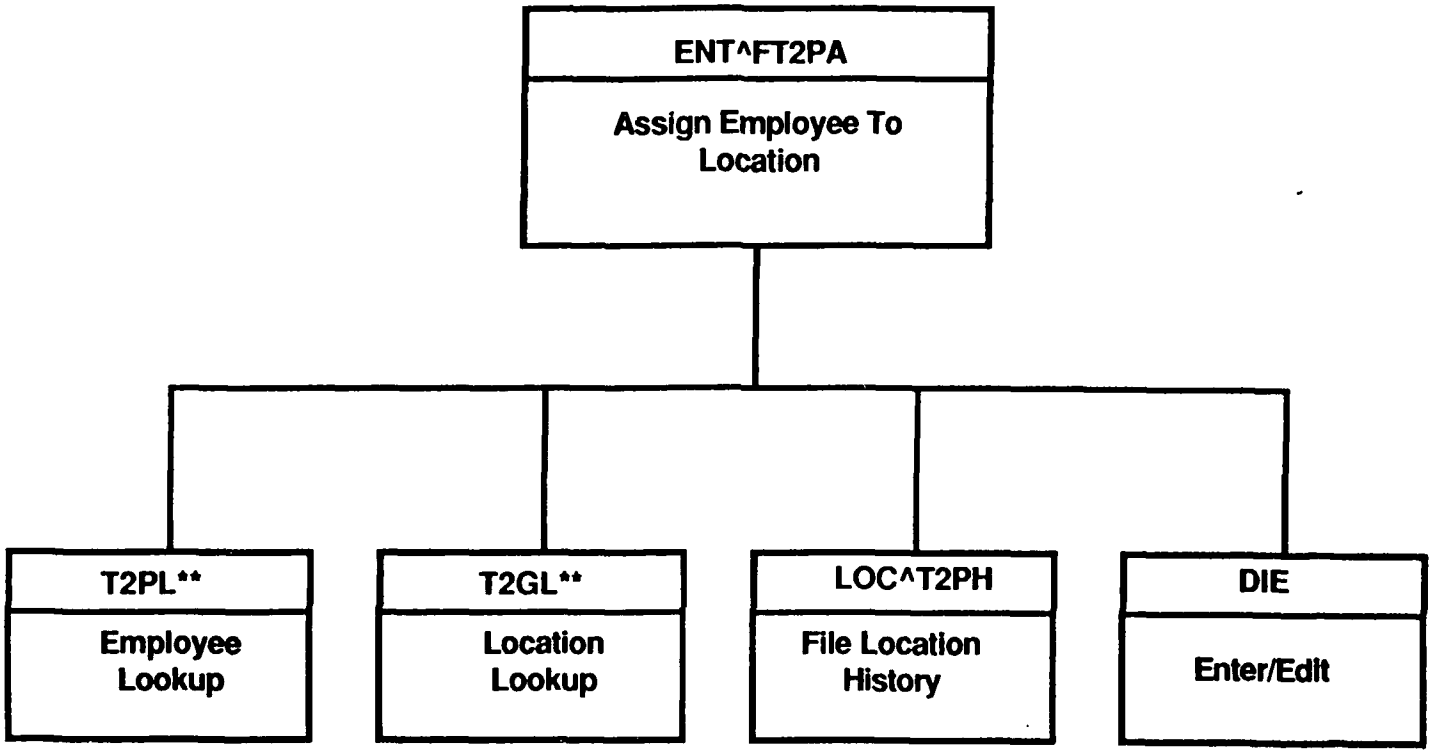

FIGURE 6-2
ASSIGN EMPLOYEE TO LOCATION OPTION
ROUTINE STRUCTURE

6.3.4 Variables

The following variables are used in this option:

- LN: Entry number of the current entry in the Current Location subfile of the Employee file and the Location History subfile
- PL: Entry number and first piece of the zero node for the current entry in the Location file
- PN: Entry number of the current entry in the Employee file
- PS(0): The date on which the location was assigned to the employee
- PS(1): The hours per week the employee spends at the current location
- PS(2): The termination status; set to "T" if the location has been terminated or removed for that employee
- PY0: The zero node of the current entry in the Current Location subfile

6.4 Inactivate Location

6.4.1 Purpose

This option updates the Location file to indicate that a location is no longer to be used.

6.4.2 Overview

This edit option, illustrated in Figure 6-3, calls routine T2GLI at tag ENT. This routine calls T2GL to prompt the user for a location (see Section 9.6). Then a lookup is done using DIC to determine whether there are subordinated locations active for the selected location. If so, the user may not inactivate the location. Similarly, if there are personnel, agency units, or sampling goals assigned to the location, it may not be inactivated. If none of these conditions is true, the user is prompted to choose whether to continue with the inactivation. If he/she so chooses, routine T2GED is used to prompt for an inactivate date.

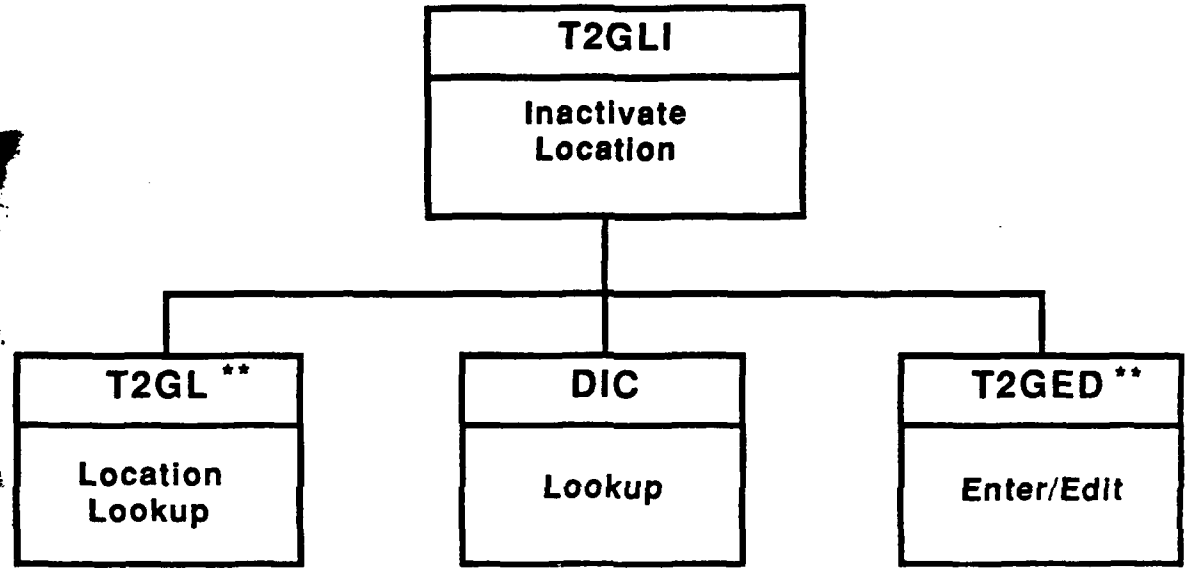

FIGURE 6-3
INACTIVATE LOCATION OPTION
ROUTINE STRUCTURE

6.4.3 Globals Referenced

The following globals are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Location	---	1073	↑AGENCY(1073,	ADMIN	Update
Employee	---	1004	↑EMPLOY(	ADMIN	Read
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Sampling Goals	---	1117	↑EXP(1117,	EE	Read

6.4.4 Variables

The following variables are used in this option:

- GNM: A Location file entry
- GSTOP: A flag to indicate whether subordinate locations exist for the selected location
- GY: The entry number of the current location in the Location file

7.0 PERSONNEL PROCESSES

7.1 Introduction

The Employee options of the Administration module provide the System Manager with the means to maintain the Employee file. Under normal operations, the Employee file will be loaded from data tapes provided from the Naval Civilian Personnel Data System (NCPDS). This load process will add newly hired employees to the Employee file, modify (edit) data values within a person's Employee file entry, transfer an employee from one agency unit to another or from one occupation to another, and record an employee termination.

Additional functions are also provided so that any of the employee transactions may be entered manually. The ability to enter an employee into the system prior to his or her actual employment enables authorized users to create an Employee file entry for someone who is being evaluated for an occupation that has medical requirements (a Prehire person). The system then automatically links the employee to the medical programs required by the entered occupation. The clinic can use this information to conduct the required examinations. Another manual feature provided by the Employee options allows authorized users to create an Employee file entry for someone who is not in the data base, but who is filing a compensation claim (a Comp Only person). This section describes the manual and NCPDS options by which the Employee file is maintained.

7.2 Enter/Edit Employee

7.2.1 Purpose

The Enter/Edit Employee option is used to create a new entry in the Employee file. Since most entries into the Employee file will occur as a result of loading data tapes from personnel systems, this option primarily serves to provide a manual entry process by which one may enter Prehire employees who are not found on the data tapes. It also serves as a means for manually modifying the data on file for an employee. This option does not process Compensation Only employees.

7.2.2 Overview

The routine T2PEN is used for the Enter/Edit Employee option. Figure 7-1 presents the routine structure. First the routine unlocks all files, and then sets the date variable to the current date, if it is not already set. T2PL is used to look up employees. If the employee selected is already on file, the routine branches to the EDIT line tag. For new employees, who were not found in the T2PL lookup, DIC is called to add the

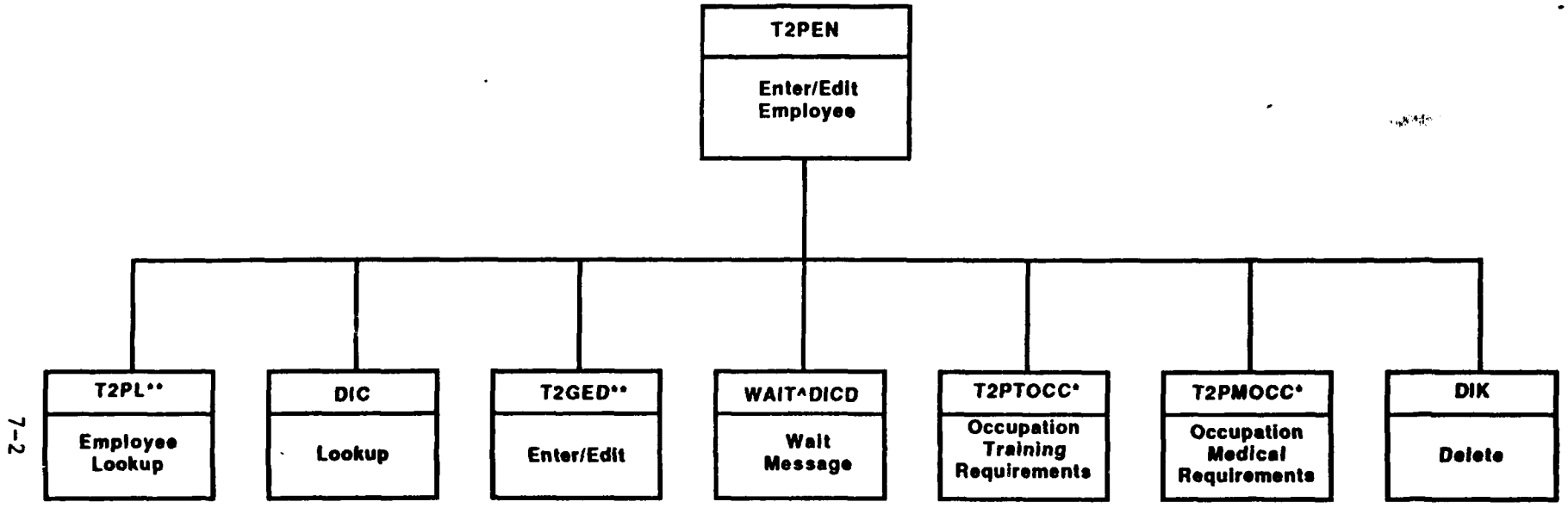

FIGURE 7-1
ENTER/EDIT EMPLOYEE OPTION
ROUTINE STRUCTURE

employee to the file. The DIC("DR") string stuffs the Create Date field with the date variable for the current date and prompts for the Agency field. Beginning at the line tag OLD, the file entry is locked and the user is prompted to choose whether the entry being made is for a Prehire or a New Hire. If neither of these is chosen, the routine deletes the newly created entry using paragraph DEL and loops to the ENT line tag to begin the sequence over with the selection of a new employee.

For Prehire employees, the routine uses the ↑AGENCY global to determine which of the agencies allow Prehires to be entered into the Employee file. If the user has chosen to enter an employee into an agency that does not have a Prehire agency unit, the Employee file entry is deleted, using paragraph DEL and the entry process is begun again. T2GED is used to control the input of Prehire data for an employee. If a new occupation has been entered for an employee, or if a new employee is entered, the routine goes to the tag OCC, where routine DICD is called to warn the user that the system will pause for a while and T2PTOCC and T2PMOCC are used to file the training and medical requirements.

For New Hires, the process continues at tag HIRE by setting up the variables for the T2GED routine. T2GED is called; paragraph OCC is performed if the Occupation field has been changed or if the employee is a new entry.

If the employee's entry is being edited, the input template EMPLOYEE is used to drive the prompt sequence. As in the above cases, if the Occupation field has been edited, OCC is performed.

7.2.3 Globals Referenced

The following globals are read or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Employee	---	1004	↑EMPLOY(	ADMIN	Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Supervisory Level	---	1135	↑AGENCY(1135,	ADMIN	Read
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read
State	---	1003	↑DIZ(1003,	ADMIN	Read
Course Catalog	---	1113	↑TRN(1113,	S/HT	Read
Medical Program	---	1088	↑MED(1088,	MES	Read
Reason for Medical Visit/ Exam	---	1128	↑MED(1128,	MES	Read

7.2.4 Variables

The following variables are used in this option:

- JLIM: Agency pointer value of employee entry.
- MP: Employee file pointer value.
- PAG: Agency pointer value of employee entry.
- PAGU: Agency Unit pointer value of employee entry.
- PCOMP: Existence of this variable indicates that the routine is being entered through the Enter/Edit Compensation Only Employees option.
- PHDT: Hire Date in FileMan format.
- PLOAD: A flag to indicate if the entry of the employee is being done via tape or manually. $D(PLOAD) indicates the data is coming from tape, '$D(PLOAD) is for manual entry.
- PN: Employee file pointer value.
- PNEW: Set to equal the employee's name surrounded by double quotes to ensure creation of a new entry for an employee name that was not selected, but passes the name edits.
- PON: Newly-entered Occupation pointer value.
- POO: Old Occupation pointer value.
- PPRE: Agency Unit pointer for "Prehire" agency Unit within the selected agency.
- PY0: Zero node of selected employee's entry.
- TEFF: Effective Date of Occupation (used in T2PTOCC).
- TEMP: Employee file pointer value (used in T2PTOCC).
- TOCC: Occupation pointer value (used in T2PTOCC).

7.2.5 Remarks

This option does not set the PCOMP variable. See Section 7.8 for a description of how T2PEN works when this variables is set.

The Job Title field is stuffed with the name of the Occupation by a trigger cross reference on the Occupation field.

When a Hire Date is filed, the Terminate Date field is automatically cleared (set to null) by a trigger. Terminate Date triggers the filing of the Hire History using HIS↑T2PTERM. In the case of this option, since the Terminate Date value is "@" (for deletion) HIS↑T2PTERM is quit immediately, and no filing is done.

Triggers on the Occupation Effective Date and Agency Unit Effective Date field cause the Occupation History and Agency History to be filed using OCC↑T2PH and AGENCY↑T2PH respectively.

The Internal Name field is triggered from the entry of the Employee Name after T2PS strips the entered name of punctuation and extraneous blanks. If the stripped name is contained in the name as entered, the Internal Name field is not filed. The variable in which the stripped name is returned from T2PS is IN.

Section 9.11 discusses the Occupation Medical/Training Requirements utility routines T2PMOCC and T2PTOCC.

7.3 Terminate Employee

7.3.1 Purpose

This option is used to record the termination of an employee's employment and to remove the employee from medical exam programs, training programs, and locations.

7.3.2 Overview

This edit option, illustrated in Figure 7-2, calls routine T2PTERM. This routine first checks the Occupation file to determine whether terminated employees are allowed in the system. If they are, routine T2PL (see Section 8.3) is called to prompt the user for an employee name. The employee's agency is checked to determine whether employees can be terminated in this agency. If so, the termination date is entered into the Employee file for that employee using routine DIE. After DICD is called to warn the user of a system pause, the occupation "terminated" and the agency name are entered into the Employee file by routine DIE. Routine T2PH is then called at line tag LOC to file the termination in the Location History subfile of the Employee file, routine DIK deletes the location from the Current Location subfile, the Enrollment Removal utility is called to remove the employee from all medical programs, and all current training requirements are removed from the employee's entry in the Current Course subfile of the Employee file using routine DIK. When the termination date is entered into the Employee file, the instructions at line tag HIS in T2PTERM are executed. These instructions record the hire date and termination date in the Hire History subfile of the Employee file.

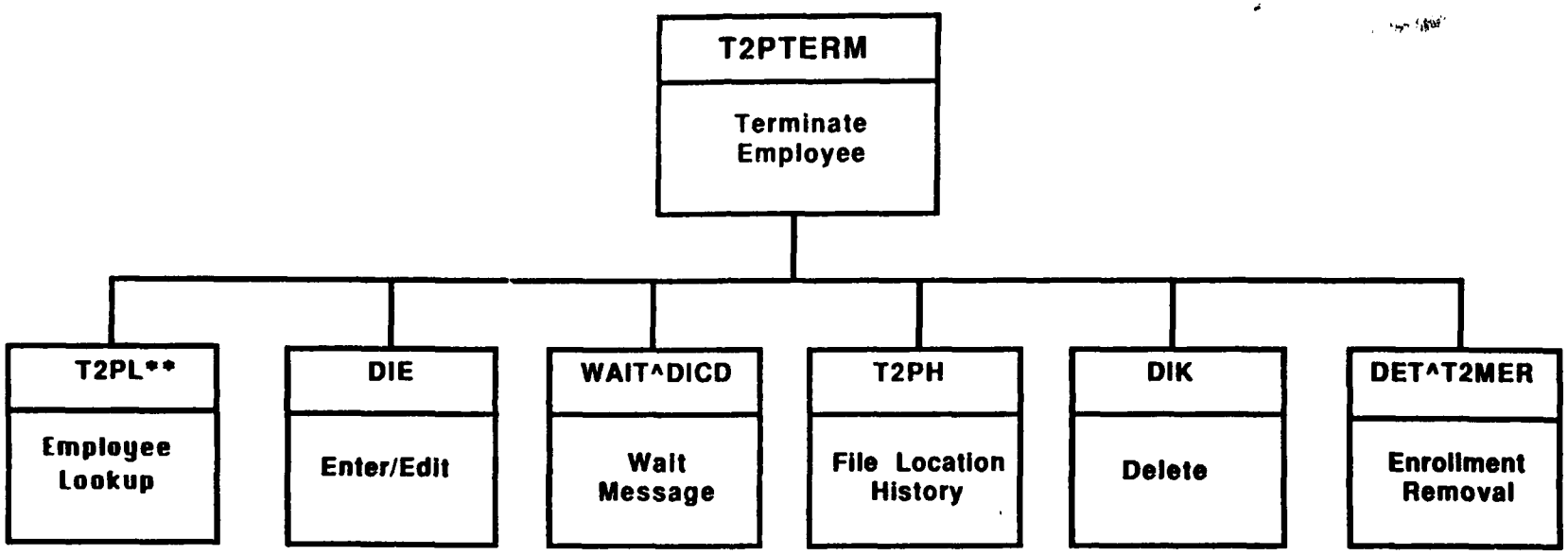

FIGURE 7-2
TERMINATE EMPLOYEE OPTION
ROUTINE STRUCTURE

Routine T2PH is a utility routine that updates several history subfiles in the Employee file. When the routine is called at the tag LOC, the routine updates the Location History subfile. First it forces a new entry with a call to DIE, using a DR string containing the .01 field. Then it updates the entry with an effective date, hours per week, and termination status.

See the Medical Exam Scheduling Module Program Maintenance Manual, Section 9.3, for a discussion of the Enrollment Removal utility.

7.3.3 Globals Referenced

The following files are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Employee	---	1004	↑EMPLOY(	ADMIN	Update
Employee	Current Location	1004.03	↑EMPLOY(#,4,	ADMIN	Update
Employee	Location History	1004.04	↑EMPLOY(#,5,	ADMIN	Update
Employee	Medical Program	1004.05	↑EMPLOY(#,3,	ADMIN	Update
Employee	Current Course	1004.11	↑EMPLOY(#,10,	ADMIN	Update
Employee	Hire History	1004.01	↑EMPLOY(#,6,	ADMIN	Update

7.3.4 Variables

The following variables are used in this option:

- JLIM: The entry number in the Agency Unit file for the employee's agency
- LN: Entry number of the current location in the Current Location subfile and the Location History subfile (same as PDA)
- PDA: The entry number of the current entry in the Current Location subfile of the Employee file
- PDT: The termination date used when data comes in on tape
- PHD: Hire date for the current employee

- PLN: Pointer to the Location file for the current location
- PLOAD: A flag that is set if data is coming in on tape; the flag is not set if the user is manually entering data
- PMSG: An array containing error messages
- PN: The entry number in the Employee file for the current employee
- PS(0): Effective date of removal of employee from a location
- PS(1): Hours an employee works in a location
- PS(2): Termination status for an employee in a location
- PTAG: The entry number in the Agency Unit file for the employee's agency
- PTD: Termination date for the current employee
- PTOCC: The entry number in the Occupation file for the occupation "terminated"
- XDAT: The employee's termination date

7.4 Transfer Employee (Shop)

7.4.1 Purpose

This option edits the agency unit field in the Employee file.

7.4.2 Overview

This edit option, illustrated in Figure 7-3, calls routine T2PTRA at the line tag ENT. First T2PTRA calls routine T2PL to do a lookup on an employee (see Section 9.3). Then it checks to see that there is a hire date and no termination date for the selected employee before continuing at line tag EDIT. It calls routine T2GED and uses a DR string to edit the agency unit field and agency unit effective date field.

7.4.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Employee	---	1004	↑EMPLOY(	ADMIN	Update

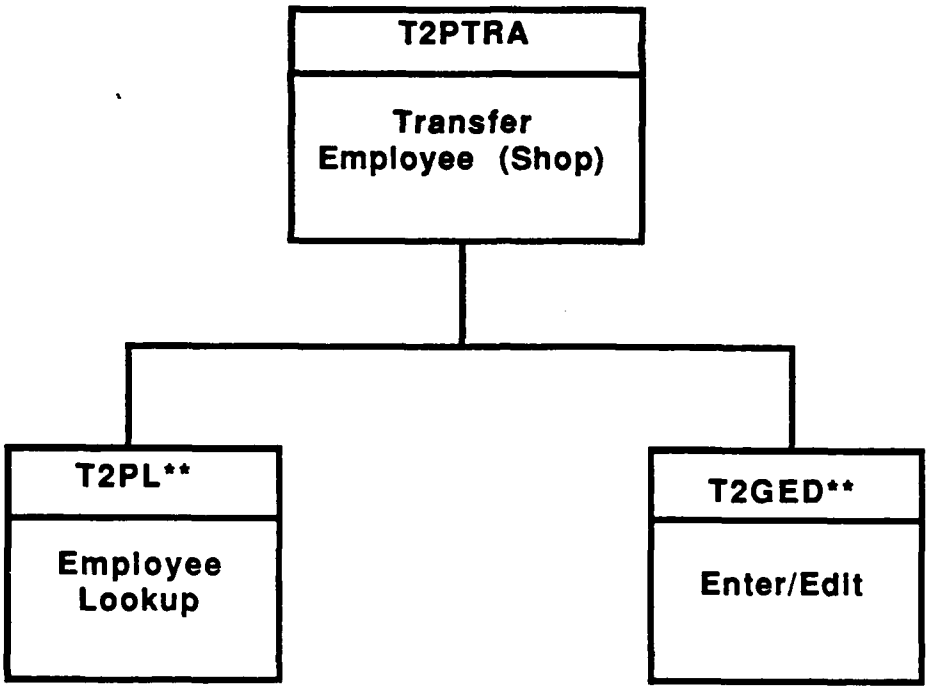

FIGURE 7-3
TRANSFER EMPLOYEE (SHOP) OPTION
ROUTINE STRUCTURE

7.4.4 Variables

The following variables are used in this option:

- JLIM: Agency field of current entry in Employee file
- PN: Entry number of current employee in Employee file
- PYO: Zero node of current entry in Employee file

7.5 Display Employee

7.5.1 Purpose

This option displays an entry in the Employee file.

7.5.2 Overview

This option calls routine T2PR3. Routine T2PL is used to select an employee (see Section 9.3) and then DIP is called at the line tag EN1 using print template EMPLOYEE DISPLAY to format output. Figure 7-4 shows the routine structure.

7.5.3 Globals Referenced

The following globals are read in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Employee	---	1004	↑EMPLOY(	ADMIN	Read

7.5.4 Variables

The following variable is used in this option:

- PAG: Agency of employee in Employee file

7.6 List Employees by Agency Unit

7.6.1 Purpose

This option produces a report of employees for a selected agency unit. The user can also limit the report to employees in a given termination date range.

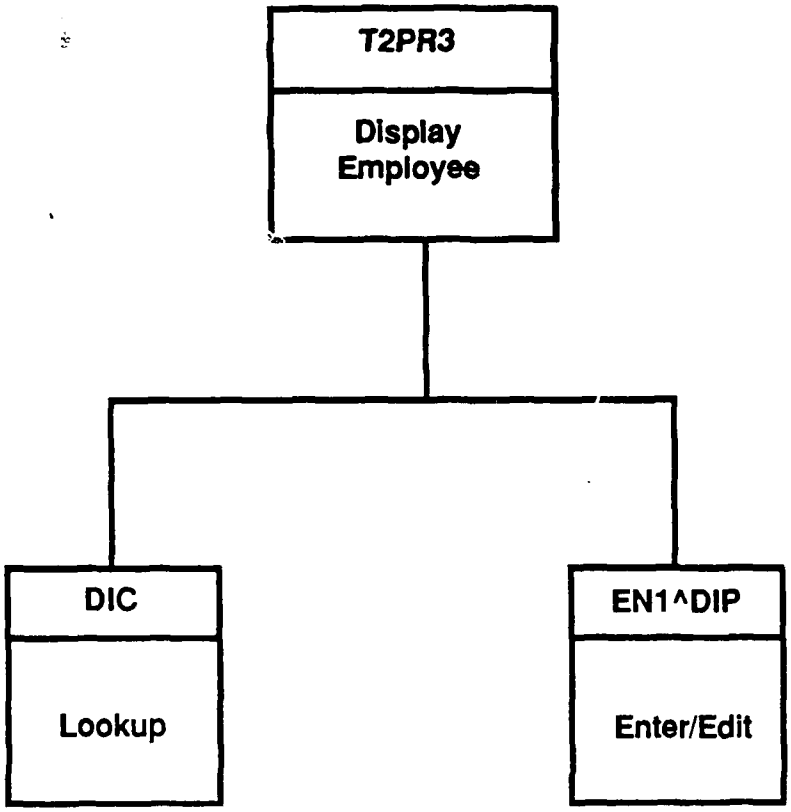

FIGURE 7-4
DISPLAY EMPLOYEE OPTION
ROUTINE STRUCTURE

7.6.2 Overview

This print option calls routine T2PR2. The user is prompted to select an agency by a call to DIC. Then special sort variables (JSTR, JWITH) are set. If the report is for TERM, COMP, or PREHIRE employees, the report is printed with the template SPECIAL EMPLOYEES. All other employee reports are printed with the template CURRENT EMP IN AGENCY UNIT. All reports are printed by a call to DIP at the tag EN1. Figure 7-5 shows the routine structure.

7.6.3 Globals Referenced

The following globals are read in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Employee	---	1004	↑EMPLOY(	ADMIN	Read

7.6.4 Variables

- JAG: Code of the agency to which the agency unit belongs
- JAGN: Agency code of the selected agency unit concatenated with the agency unit name
- JSTR: Astring of the selected agency unit
- JWITH: A flag indicating whether the report should include descendants of the selected agency unit
- JYO: Zero node of the selected agency unit in the Agency Unit file

7.7 List Employees by Location

7.7.1 Purpose

This option produces a report of employees sorted by location and name.

7.7.2 Overview

This print option calls routine T2PR5. The user is prompted to select a location. If "all" locations are selected, the "from" and "to" nodes of the DIBT global are set for all locations and control passes to

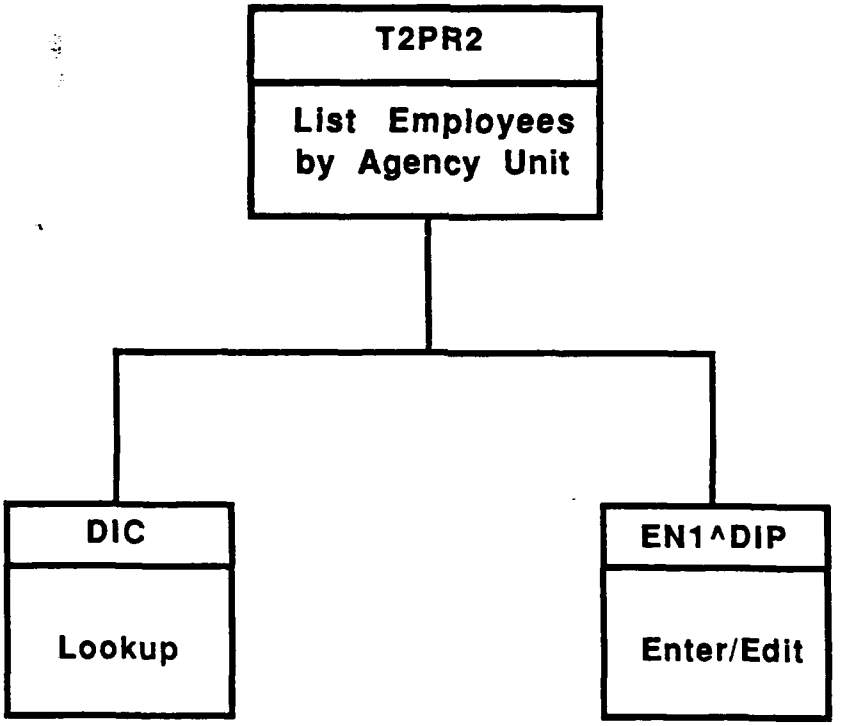

FIGURE 7-5
LIST EMPLOYEES BY AGENCY UNIT OPTION
ROUTINE STRUCTURE

tag GO. Otherwise routine T2GL is called at the line tag EN1 for a lookup on location (see Section 9.6). Then the "from" and "to" nodes of the DIBT global are set. The report is printed by routine DIP at the line tag EN1, using the print template EMP BY LOCATION and the sort template T2EL. The routine structure is illustrated in Figure 7-6.

7.7.3 Globals Referenced

The following globals are read in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Sort Template	---	.401	↑DIBT(	---	Read
Employee	---	1004	↑EMPLOY(	ADMIN	Read

7.7.4 Variables

The following variable is used in this option:

- AST: Entry number of sort template T2EL in global DIBT

7.8 Enter/Edit Compensation Only Employee

7.8.1 Purpose

This option is used by the Compensation Office to create or edit an Employee file entry for a person who is filing a claim for compensation. Any existing employee entry can be edited even if it is not a Compensation Only entry. If an existing entry is not found, a Compensation Only entry can be created. The Employee file must contain an entry for each person who submits claims to be managed by the Injury and Compensation Claims module (ICC). The Employee file contains entries for employees who are currently working in the local facility or who have been active employees in the facility since the installation of the system. Claims are sometimes submitted for employees who have terminated their employment prior to the installation of the system, or who were employees by another facility whose claims are managed in the local Compensation Office. In these cases, the option to create an entry for the employee is necessary. An Employee file entry created through this option is flagged as "Compensation Only", and it may not be accessed by modules other than ICC unless the person becomes an active employee.

7.8.2 Overview

The routine T2PEN is entered at line tag ENTC to perform the Enter/Edit Compensation Only Employee option. Figure 7-7 presents the routine structure.

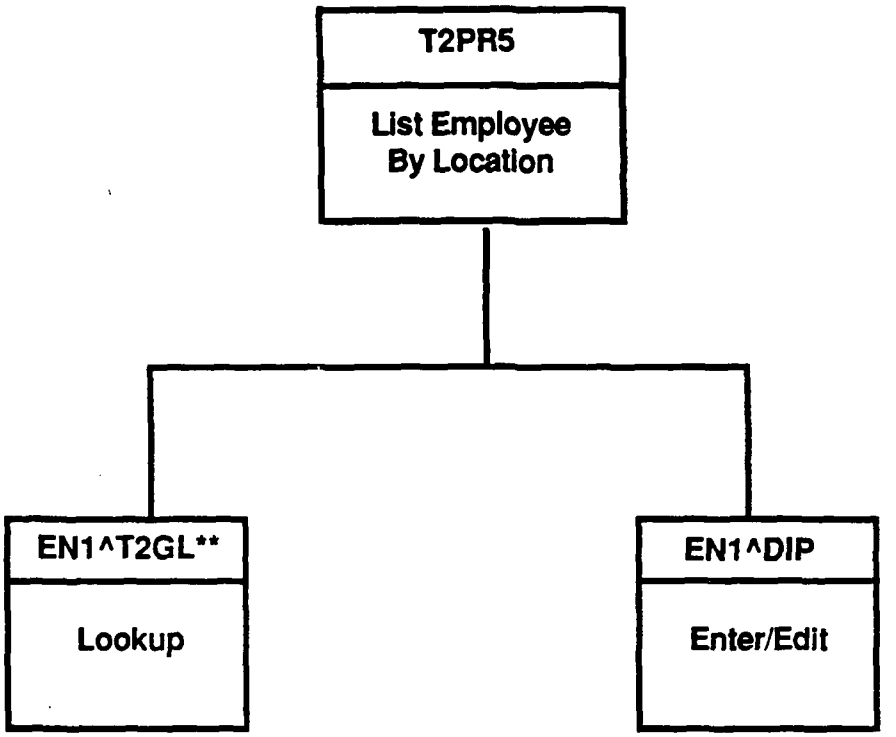

FIGURE 7-6
LIST EMPLOYEES BY LOCATION OPTION
ROUTINE STRUCTURE

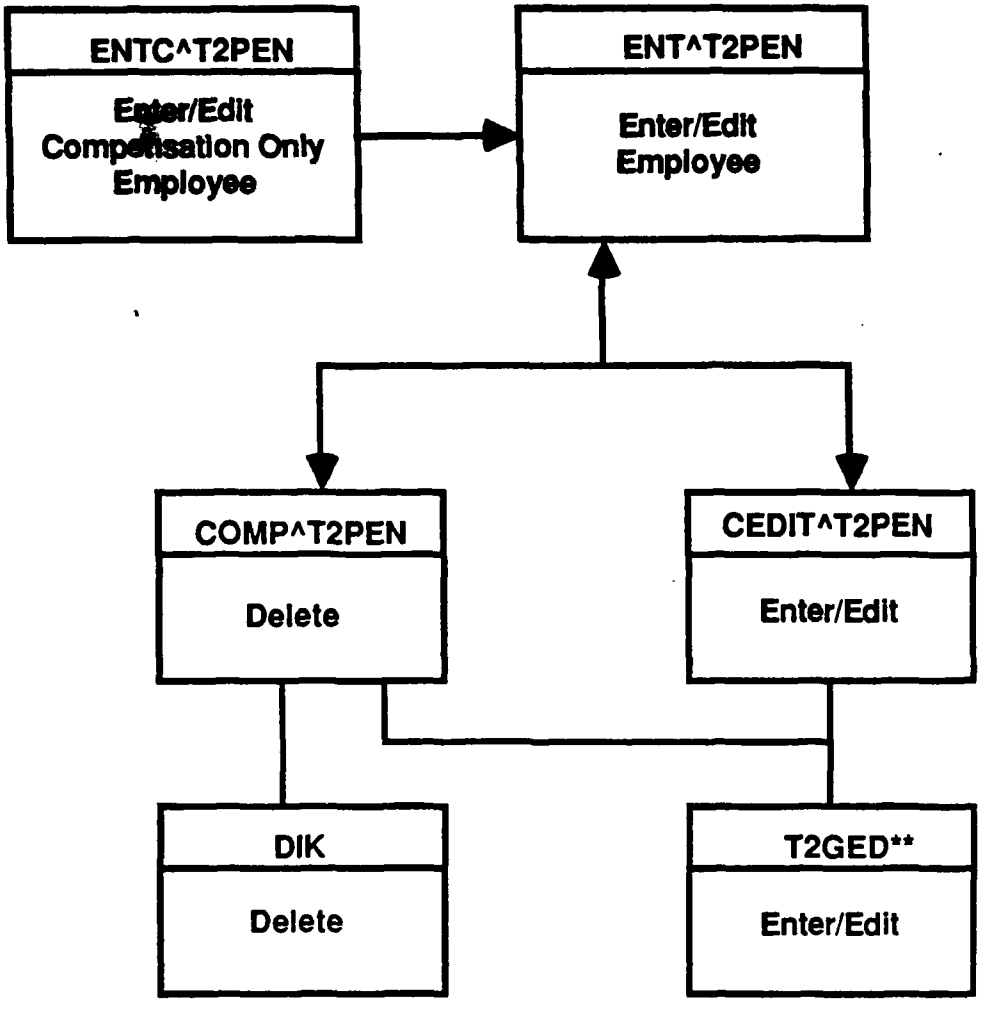

FIGURE 7-7
ENTER/EDIT COMPENSATION ONLY EMPLOYEE OPTION
ROUTINE STRUCTURE

At line tag ENTC, the variable PCOMP is set to indicate that the option being used is Enter/Edit Compensation Only Employee and control is passed to the normal entry point for routine T2PEN. Then, T2PEN unlocks all files and then sets the date variable (DT) to the FileMan value for the current date, if DT is not already set. Then T2PL is used to look up the employee (see Section 9.3). If the employee selected is already on file, the routine branches to the CEDIT tag for editing "Comp Only" employees, i.e., records for which the value of the Compensation Only field (#11) is set to "C". If the selected employee is not "Comp Only", editing will proceed as described in Section 7.2.2. If the employee is not already on file, DIC is used to create a new "Comp Only" entry. The DIC("DR") string stuffs the Create Date field with the value of DT, the Managing Agency pointer is set into the Agency field, and "C" is stuffed into the Compensation Only field (#11). At the tag OLD, the file entry is locked. To complete entry of "Comp Only" employees, the routine branches to the paragraph COMP.

At the line tag COMP, the routine checks the Agency Unit file to ensure that a "Comp Only" agency unit exists in the Managing Agency of the employee. If there is no "Comp Only" unit on file, the user is given a message, and the Employee file entry is deleted using paragraph DEL. Otherwise T2GED is used to control the data entry. In the DR string, the pointer for the "Comp Only" agency unit found in the Managing Agency is stuffed into the Agency Unit field of the Employee file entry.

7.8.3 Globals Referenced

The following globals are read or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Employee	---	1004	↑EMPLOY(	ADMIN	Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read
State	---	1003	↑DIZ(1003,	ADMIN	Read

7.8.4 Variables

The following variables are used in this option:

- JLIM: Agency pointer value of employee entry
- PAG: Agency pointer value of employee entry
- PAGU: Agency Unit pointer value of employee entry
- PCOMP: Initially, a flag to indicate that the option selected is for entry of a Compensation Only employee, then becomes Agency pointer value for Managing Agency

- PHDT: Hire Date in FileMan format
- PN: Employee file pointer value
- PNEW: Set to equal the employee's name surrounded by double quotes to ensure creation of a new entry for this employee
- POO: Set to null (old value of Occupation field)
- PYO: Zero node of selected employee's entry

7.8.5 Remarks

The Job Title field is stuffed with the name of the Occupation by a trigger cross-reference on the Occupation field.

Triggers on the Occupation Effective Date and Agency Unit Effective Date field cause the Occupation History and Agency History to be filed using OCC↑T2PH and AGENCY↑T2PH respectively.

The Internal Name field is triggered from the entry of the Employee Name after T2PS strips the entered name of punctuation and extraneous blanks (see Section 9.12). If the stripped name is contained in the name as entered, the Internal Name field is not filed. The variable in which the stripped name is returned from T2PS is IN.

7.9 Load NCPDS Transaction File

7.9.1 Purpose

This option processes an NCPDS tape. Each entry is filed into the NCPDS Transactions file as of the tape date. Any fatal errors are flagged in the transaction entry. Fields with non-fatal errors are changed to null.

7.9.2 Overview

This option is performed by routine T2PUTL. Figure 7-8 illustrates the routine structure. After locking the transaction file, the routine sets the error trapping variable which is used to recognize errors and a valid end-of-file. Next, the tape drive is opened on device 47 and the first record is read and verified as a valid transaction date using routine %DT. If there are transactions on file for that date, line ASK verifies that the user wants to continue processing. Then, NCPDS Edits file entries are reviewed to ensure that they make sense before processing begins, including verification that there is an "SSNDT" entry for handling the social security number (SSN) date.

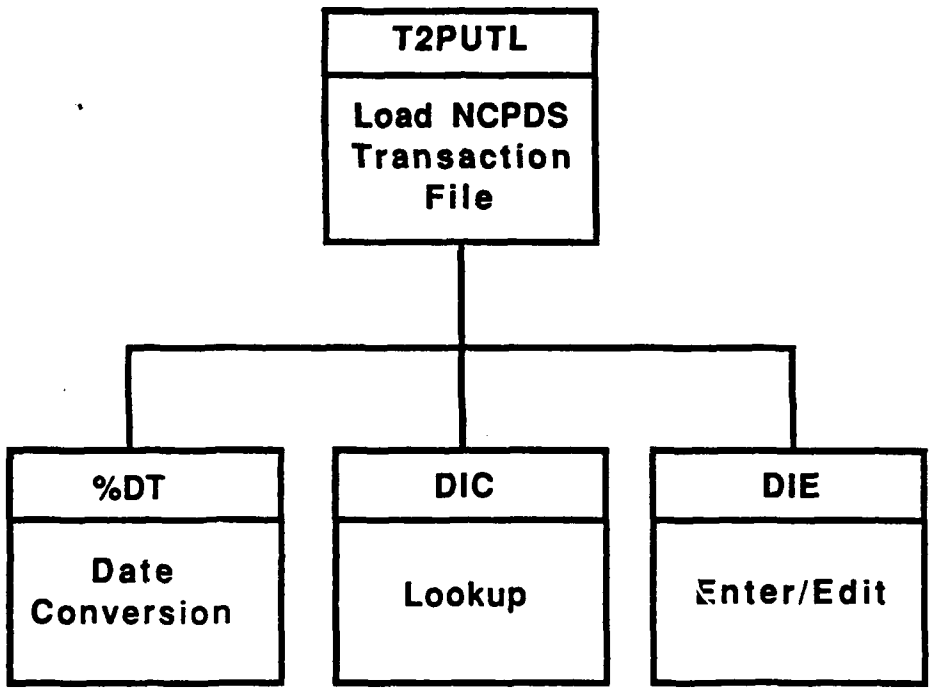

FIGURE 7-8
LOAD NCPDS TRANSACTION FILE OPTION ROUTINE STRUCTURE

Starting at line READ, a tape record is read and then filed into the transaction file at the next available transaction number using routine DIC. The transaction date is stuffed into the entry using routine DIE. Then, the tape fields are processed in order of their start column as designated by the edit file. The original tape version of the field is stuffed into the appropriate transaction file field using routine DIE. For null entries that are not for a required field, control is passed back to line EDIT to get the next field. For required fields that are null, control goes to line ERR which will set the appropriate flag. Non-null fields are edited using the edit file Translation Code field which will change an invalid entry to null. For a field that is null as a result of the edit, control is passed to line ERR, which sets a flag if the field is required. Fields that pass the first edit are subject to the transaction file edit for the field, which will kill an invalid entry. Fields that pass all edits are stuffed into the appropriate field at line OK using routine DIE. For fields that fail the final edit, control is passed to line ERR. In addition to setting the error flag for a required field, a field value that passed the first edit but not the second will be stored in the transaction file entry using the same call to routine DIE.

On an error or on reading an end-of-file, control is passed to line TERR. If an end-of-file was found, processing is completed by passing control to line CL. On an error, all transactions filed with the same transaction date are deleted using routine DIK, including any that existed before the start of the run. After closing the tape unit, the routine will list and delete the cross references showing UICs deleted for the transaction date. Then, for successful runs, the UICs covered by the transaction date are listed and the special cross references are set.

Line TEST is an entry point that can be used to test the routine from programmer mode.

7.9.3 Globals Referenced

The following globals are read and/or updated:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
NCPDS Transactions	---	2000	↑PTRANS(	ADMIN	Update
NCPDS Edits	---	2001	↑DIZ(2001,	ADMIN	Read
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read
Supervisory Level	---	1135	↑AGENCY(1135,	ADMIN	Read

7.9.4 Variables

- $ZA: Indicates the outcome of a read; used to check for an end-of-file
- $ZT: Indicates the line and routine to which control is passed when an error occurs
- TDATE: The tape transaction date in FileMan format
- XE: The Ends in Column value for an edit file entry
- XED: The Ends in Column value for the SSNDT edit file entry
- XIN: (1) A tape record
  (2) The edit file entry number for SSNDT
  (3) The zero node for the edit file SSNDT entry
- XN: The zero node of an edit file entry
- XS: The Starting Column field value for an edit file entry
- XSD: The Starting Column value for the SSNDT edit file entry
- XT: An edit file entry number
- XU: (1) An agency UIC
  (2) A tape entry UIC
- XV: A tape field value
- XY: (1) A tape entry SSN Date in FileMan format
  (2) A field number for the tape version of the transaction field being processed

7.9.5 Remarks

Changes in required fields, additions of fields, or modifications to tape format can be handled by changing the NCPDS Edits file and modifying the NCPDS Transactions file; they do not require changes to this routine.

This is the only option that will file a field value even if there is no corresponding entry in the referenced file. The assumption was made that an Occupation, Agency Unit, or UIC value might be valid and not yet be in the Occupation or Agency Unit file. Keeping the "invalid" value allows users simply to add the value to the proper file before the update to make the transaction file entries valid.

The $ZT and $ZA variables are system dependent non-standard variables.

7.10 Print Transactions

7.10.1 Purpose

This option allows the user to print all or selected entries from the NCPDS Transactions file.

7.10.2 Overview

Routine T2PUTP performs this option. The routine structure is presented in Figure 7-9. Routine %DT is used to verify a user-entered Transaction Date. After getting the date, the routine asks the user for any limitations on entries based on their error status. Next, routine DIC is used to verify a user-entered UIC, with UIC selection limited to a UIC that has a valid agency entry. Finally, the requested data is printed using routine DIP and print template TAPE TRANSACTIONS.

7.10.3 Globals Referenced

The following globals are read:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
NCPDS Transactions	---	2000	↑PTRANS(	ADMIN	Read
Supervisory Level	---	1135	↑AGENCY(1135,	ADMIN	Read

7.10.4 Variables

- PDT1: The report-starting transaction date in FileMan format
- PDT2: The report-ending transaction date in FileMan format
- PERR1: The report-starting Fatal Error Flag value
- PERR2: The report-ending Fatal Error Flag value
- PUIC1: The report-starting Employee UIC value
- PUIC2: The report-ending Employee UIC value

7.10.5 Remarks

The addition or deletion of fields from the NCPDS Transactions file will require modification of the print template.

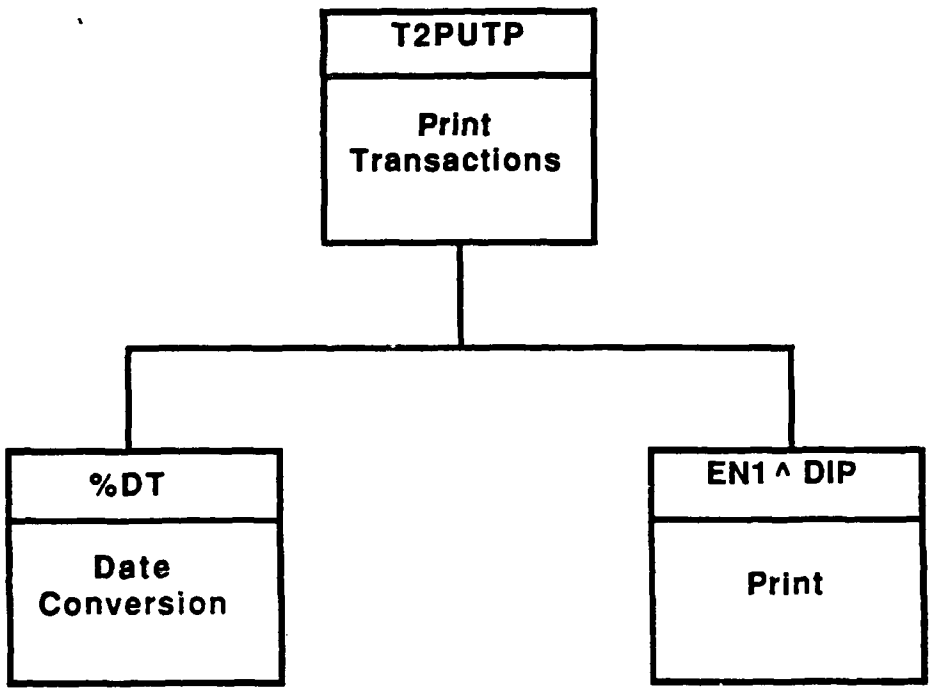

FIGURE 7-9
PRINT TRANSACTIONS OPTION
ROUTINE STRUCTURE

7.11 Edit NCPDS Transaction File

7.11.1 Purpose

This option allows users to add or edit NCPDS Transactions file entries.

7.11.2 Overview

This option is performed by routine T2PUTE. Figure 7-10 illustrates the routine structure. Routine DIC is called to select an existing entry or to create a new one for a user-specified transaction date. The entry is locked and then edited using routine T2GED and the TRANS EDIT input template. If the entry is new, an error flag is set for any required field that is missing. The user is warned if an entry has fatal error flags set.

7.11.3 Globals Referenced

The following globals are read and/or updated:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
NCPDS Transactions	---	2000	↑PTRANS(	ADMIN	Update
NCPDS Edits	---	2001	↑DIZ(2001,	ADMIN	Read
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read
Supervisory Level	---	1135	↑AGENCY(1135,	ADMIN	Read

7.11.4 Variables

In addition to standard FileMan and utility variables, routine T2PUTE uses the following variable:

- GDA: Used to identify missing required fields in a new transactions file entry

7.11.5 Remarks

The required fields have special MUMPS triggers that will set an error flag, if the field value is deleted, or delete an error flag, if the field value is entered.

The addition or deletion of NCPDS Transactions fields will necessitate changes in the input template.

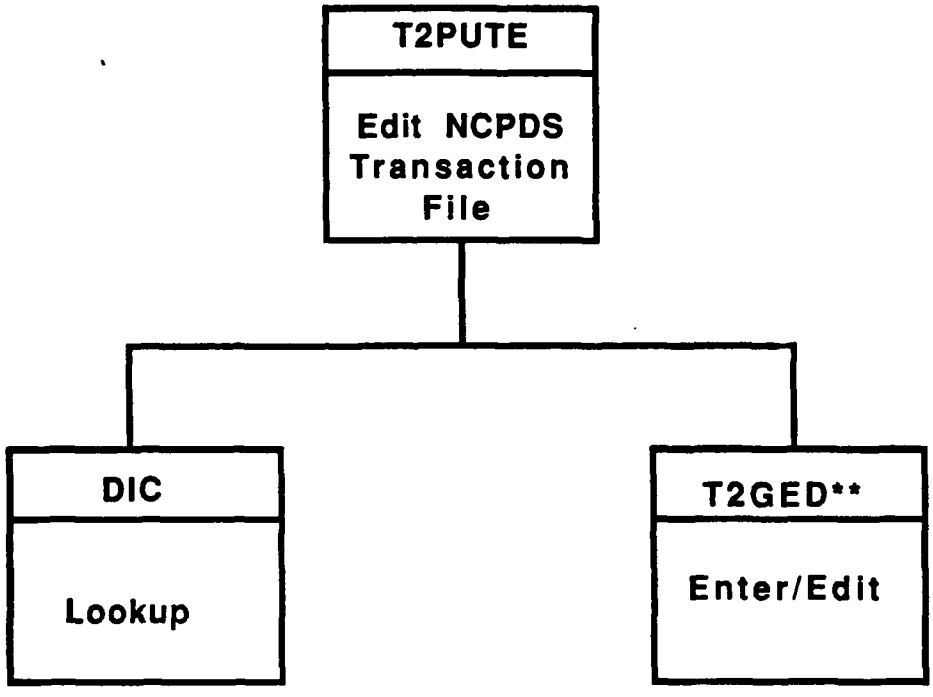

FIGURE 7-10
EDIT NCPDS TRANSACTION FILE OPTION ROUTINE STRUCTURE

7.12 Update Employee File from NCPDS Transactions

7.12.1 Purpose

This option updates the Employee file using NCPDS Transactions file entries for a given date. This includes termination of employees not having a transaction file entry, entry as new hires for transaction not in the Employee file or in the Employee file as terminated or prehire, and update of active employees. Errors and data ambiguities are listed on a report along with completed transactions.

7.12.2 Overview

Routine T2PUET performs this option. The routine structure is shown in Figure 7-11. After locking the full Employee and NCPDS Transactions files, the routine gets a transaction date and verifies it using routine %DT. If the user wants to proceed, routine %ZIS is called to select a device for the report. Next, the routine writes a report header and the transaction UICs. The routine looks through the listed UICs and the Employee file entries associated with each UIC. If there is a problem with the Employee file entry, an error message is written on the report and processing is bypassed. An Employee entry is also bypassed when it has the same UIC and SSN as a transaction file entry or when it is for a terminated employee. When there are no problems, the Employee file entry is active, and there is no matching transaction, the employee is terminated using routine T2PTERM and an entry is written on the report. Once all the employees for a UIC have been processed, the transaction file cross reference is changed to indicate that the Employee file has been processed for terminations.

After all the UICs have been processed, the routine passes control to line STP where it writes the transaction report header. Then, the transaction file entries for the selected date are processed individually. First, the UIC is checked for validity. If there is no UIC or the UIC does not pass the transaction field validity check, an error flag is set for the record by line FER, using routine DIE. Processing is bypassed if the UIC has transactions on file for a later date. If the UIC is valid, any existing error flag is killed by line KER, using routine DIK. Next, the Occupation field is checked for validity. If valid, the corresponding error flag is killed; otherwise, an error flag is set. The agency unit is checked in the same way. A social security number will be flagged as invalid if it is not in the correct format or if there is a different transaction entry with the same UIC and a matching SSN or old SSN. If the SSN is valid, the SSN error flag is killed. For entries with an old SSN, the old SSN is flagged as invalid if there is a different transaction with the same UIC and a matching SSN or an old SSN. If the old SSN is valid, the corresponding error flag is killed. If a transaction file entry has any error flags set at this point, the

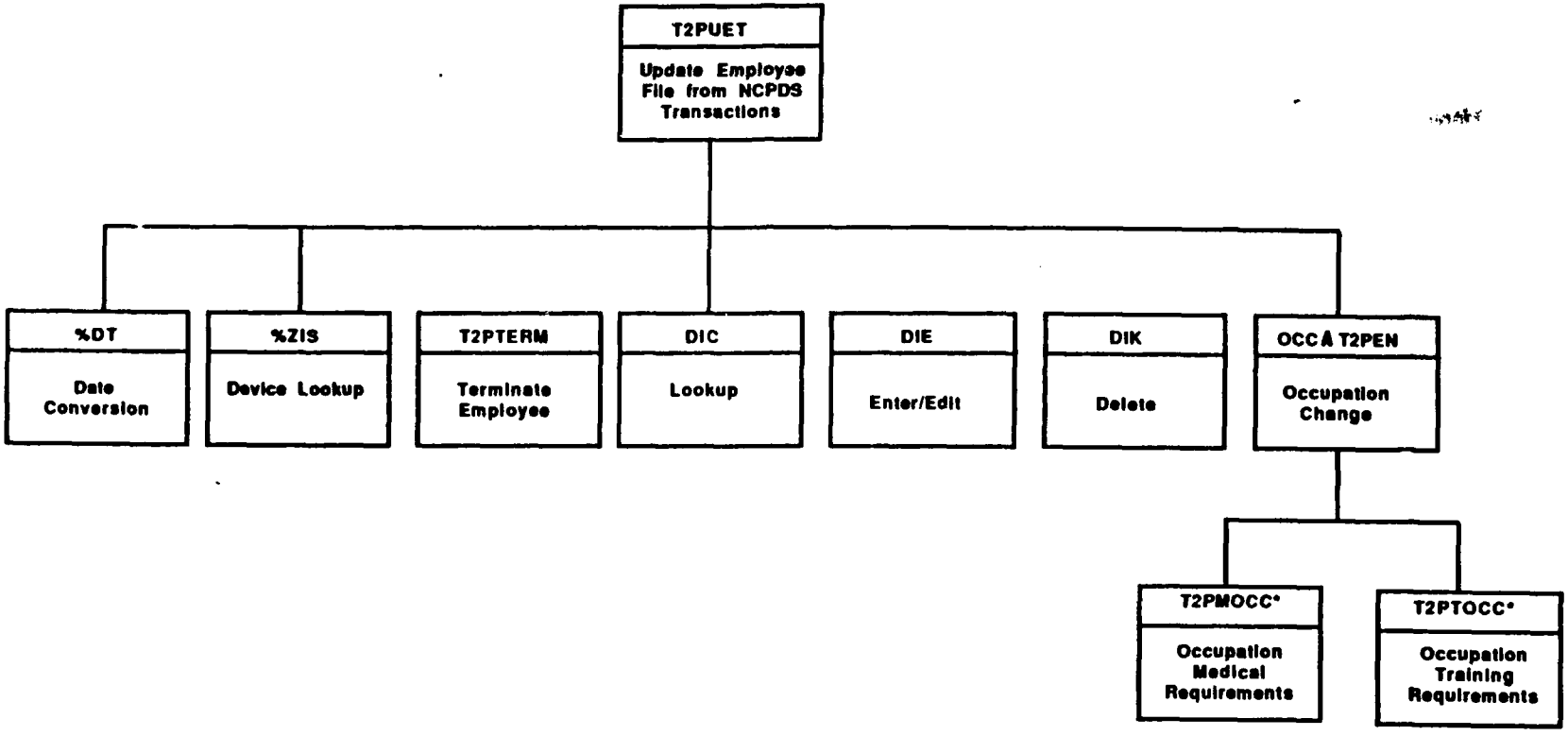

FIGURE 7-11
UPDATE EMPLOYEE FILE FROM NCPDS TRANSATIONS OPTION
ROUTINE STRUCTURE

corresponding error messages are set, the entry is printed out on the transaction report as an error, and further processing is bypassed. An entry that has no errors at this point is checked against the Employee file entries. The transaction will be printed on the transaction report as an error if there is more than one employee file match for the UIC and either the old SSN or SSN or if there are matches for the UIC and both the old SSN and SSN. When a single employee entry is found, control is passed to line OLD which identifies whether the employee entry is for an active employee or is for a prehire or termination. In either case, control is passed to line UPD.

The transaction will be added as new if there is no employee entry with an UIC and SSN that matches the transaction UIC and either the old SSN or SSN. A new Employee file entry is added by routine DIC and updated with the validated, required fields using routine DIE. If there is a problem during the update of the required fields, the new entry will be deleted using routine DIK and printed on the report as an error. Otherwise, control is passed to line UPD.

Starting at line UPD, the required fields in the transaction file entry are checked against the current Employee file entry and any changes are filed using routine DIE. IF there is a change in occupation, routine T2PEN is called at line OCC to handle the filing of training and medical requirements. For new employee entries or updates of existing prehires or terminations, the hire date is set to the provided hire date or to the transaction date using routine DIE. Next, the badge number is checked for change and filed using routine DIE. If a problem occurs while filing the badge number, an error message is set. Next, the job title is checked for changes and filed as necessary using routine DIE. Then, the remaining fields are updated using the NCPDS Edits file. For each transaction field identified, the field value is pulled and then routine DIE stuffs it into the Employee file field identified by the edit file. Once a transaction entry is processed, it is written on the report with any appropriate messages.

7.12.3 Globals Referenced

The following globals are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
NCPDS Transactions	---	2000	↑PTRANS(	ADMIN	Update
NCPDS Edits	---	2001	↑DIZ(2001,	ADMIN	Read
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read
Supervisory Level	---	1135	↑AGENCY(1135,	ADMIN	Read
Employee	---	1004	↑EMPLOY(	ADMIN	Update

7.12.4 Variables

Routine T2PUET uses the following variables in addition to standard FileMan and utility variables:

- PAD: A value of zero indicates an employee is already on file as active; a value of one indicates a new entry is being added or an existing entry is a prehire or termination that is being changed to a hire
- PAGU: A pointer to an Agency Unit file entry
- PDA: A pointer to an NCPDS Transactions file entry
- PDT: A transaction date in FileMan format
- PFN: (1) A NCPDS Edit file field number and the corresponding entry and entry number in the Error Fields subfile of the NCPDS Transactions file
  (2) An NCPDS Transactions file field number
- PLOAD: Existence of this variable notifies the utilities that user interaction is not appropriate
- PMSG: An array of error or update messages
- PN: A pointer to an Employee file entry
- PON: A pointer to an employee's new occupation
- POO: A pointer to an employee's current occupation
- PT0: The zero node of an NCPDS Transactions file entry
- PT1: The one node of an NCPDS Transactions file entry
- PTN: An NCPDS Edits file field number
- PX: A transaction UIC followed by an "X" and a SSN or old SSN; used to identify multiple occurrences of a social security number
- PXN: A transaction UIC followed by a space and the SSN; used to locate entries from the transaction "AD" cross-reference and the "AN" employee cross-reference

- PXO: A transaction UIC followed by an "X" and the old SSN; used to locate entries from the transaction "AD" cross-reference and the "AN" employee cross-reference
- PYO: The zero node for an Employee file entry
- XU: A transaction UIC

7.12.5 Remarks

This option revalidates the required fields and kills or sets flags appropriately. The NCPDS edit file is used to drive edits and updates of non-required fields. Changes in required fields or addition or deletion of update fields will require changes to this routine.

7.13 Delete NCPDS Transactions

7.13.1 Purpose

This option allows users to delete individual transactions, all transactions for a date, or all transactions for a date and UIC.

7.13.2 Overview

This option is performed by routine T2PUTD. Figure 7-12 presents the routine structure. After locking the full NCPDS Transactions file, the routine asks the user for a transaction date and verifies the entry by calling routine %DT. If the user does not specify that all transactions are to be processed, routine DIC is called to verify that an individual transaction has been specified. If a valid entry was selected, it is deleted from the file using routine DIK and the user is prompted for another transaction number.

When the user does specify that all transactions are to be processed, control is passed to line UIC. At this point, the system allows the user to specify whether processing is to be done on a given UIC, all entries, or only entries with a null UIC. Once the user has given a valid response, the system has the user verify that transactions for the specified criteria are to be deleted. If so, routine DICD is called to inform the user that processing may take some time, and routine DIK is used to delete the appropriate transactions.

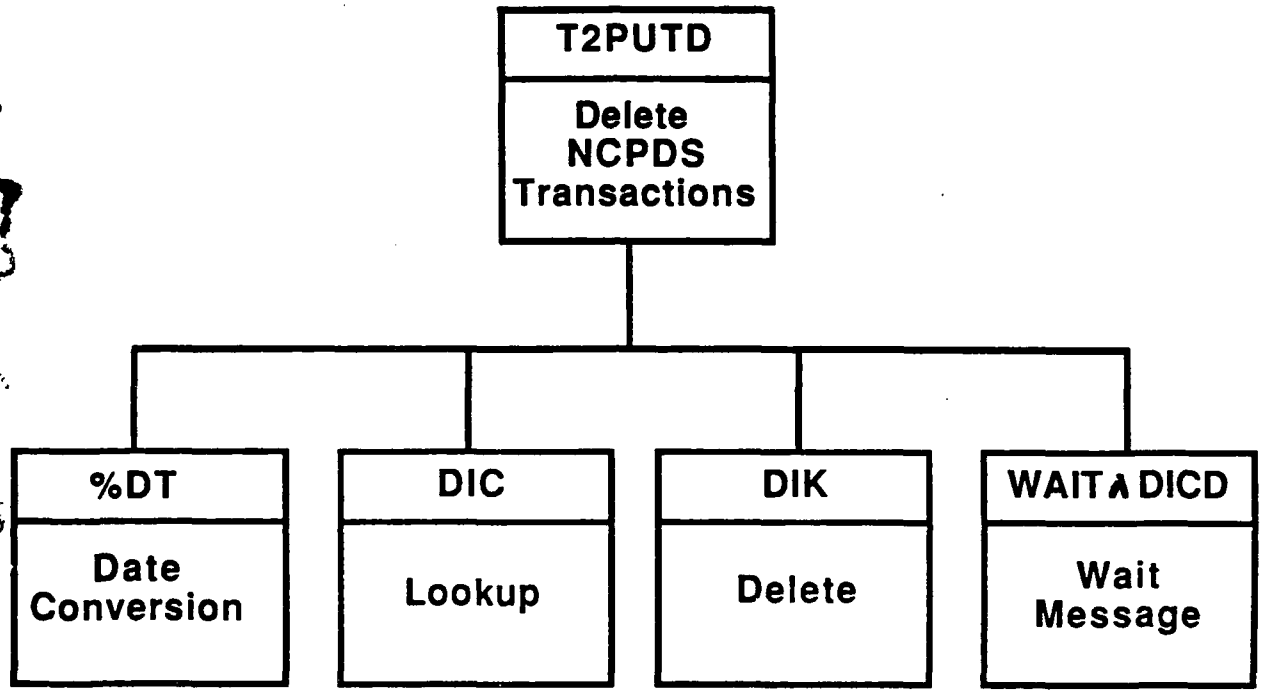

FIGURE 7-12
DELETE NCPDS TRANSACTIONS OPTION
ROUTINE STRUCTURE

7.13.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
NCPDS Transactions	---	2000	↑PTRANS(	ADMIN	Update

7.13.4 Variables

In addition to standard FileMan variables, routine T2PUTD uses the following variables:

- PDT: A transaction date in FileMan format
- PUIC: Indicates groups of transactions for deletion; contains either a single UIC, "ALL", or "NULL". The coded value "NULL" is changed to the null value

8.0 STRESSOR DATA PROCESS

8.1 Introduction

The Stressor file is used extensively within the EE module, and other modules of OSHRKS. This file contains pointers to two other files: Sample Units and Stressor Class. Sample units and stressor classifications must be entered into their own files before they can be used in creating a Stressor file entry. The remainder of this section describes the options by which the three files are maintained.

8.2 Sample Units Enter/Edit Option

8.2.1 Purpose

The Sample Units Enter/Edit option is used to maintain the vocabulary of sample units.

8.2.2 Overview

The routine T2ETUNT performs the enter/edit option for the Sample Units file. This routine calls DIC for lookup and T2GED for the entry process. Figure 8-1 presents the routine structure.

8.2.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Sample Units	---	1101	↑EXP(1101,	ADMIN	Update

8.2.4 Variables

Only standard T2GED variables are used.

8.3 Stressor Class Enter/Edit Option

8.3.1 Purpose

The Stressor Class Enter/Edit option is used to maintain the vocabulary of stressor classifications used in OSHRKS.

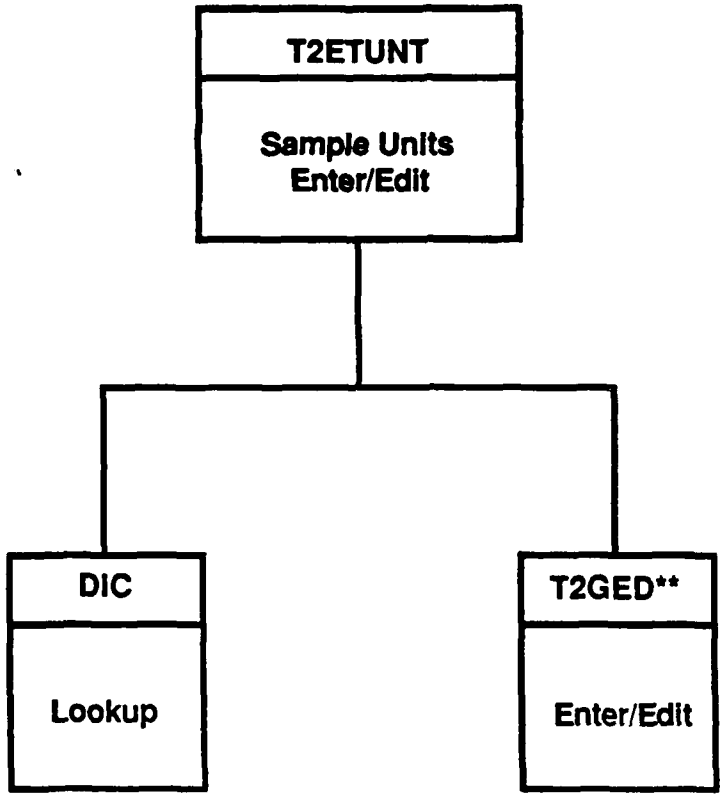

FIGURE 8-1
SAMPLE UNITS ENTER/EDIT OPTION
ROUTINE STRUCTURE

8.3.2 Overview

The routine T2ETSTR performs the enter/edit option for the Stressor Class file. This routine uses DIC for lookup and uses T2GED to perform the entry process. The routine structure is illustrated in Figure 8-2.

8.3.3 Globals Referenced

The following global is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Stressor Class	---	1039	↑STRESS(1039,	ADMIN	Update

8.3.4 Variables

Only standard T2GED variables are used.

8.4 Stressor Enter/Edit Option

8.4.1 Purpose

The Stressor Enter/Edit option is used to maintain the Stressor file. This file contains the primary name of each stressor, synonyms, CAS and NIOSH numbers, the medical monitoring rank, risk assessment category, classifications, and exposure limits. It is used extensively within the EE module and by other modules of OSHRKS as well.

8.4.2 Overview

The routine that performs the Stressor Enter/Edit option is T2SEN. This routine uses T2SL for lookup and the input template STRESSOR INPUT with DIE to control the entry of data. The routine structure is shown in Figure 8-3.

8.4.3 Globals Referenced

The following files are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Stressor	---	1083	↑STRESS(0,	ADMIN	Update
Sample Units	---	1101	↑EXP(1101,	ADMIN	Read
Stressor Class	---	1039	↑STRESS(1039,	ADMIN	Read

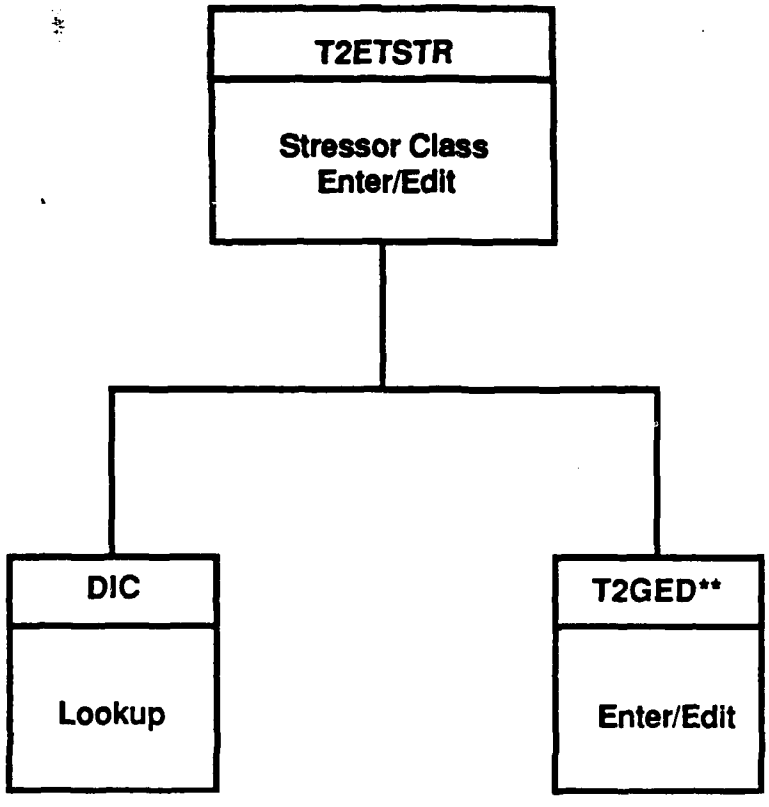

FIGURE 8-2
STRESSOR CLASS ENTER/EDIT OPTION
ROUTINE STRUCTURE

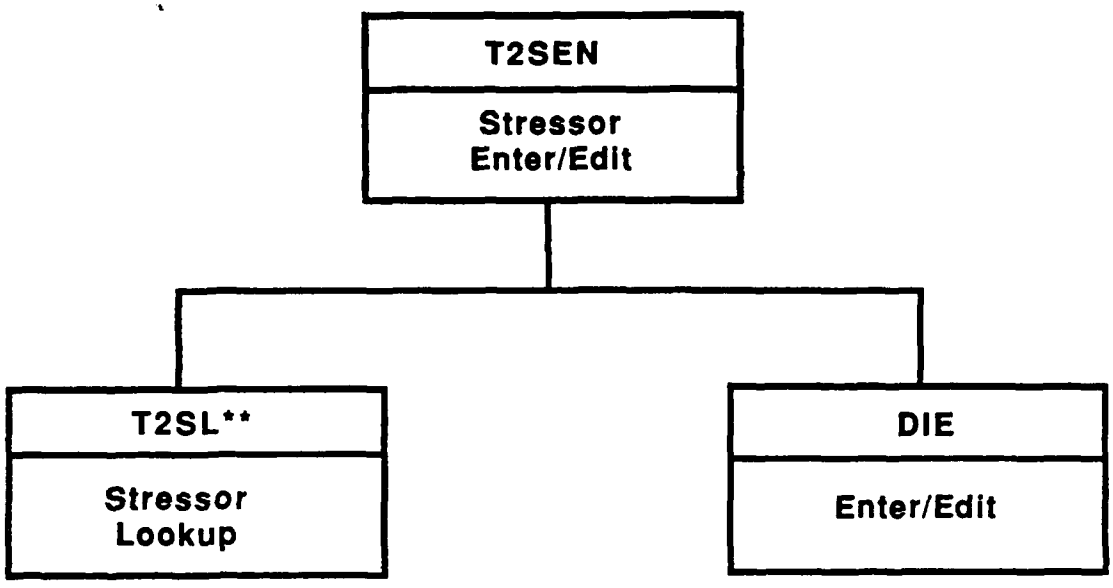

FIGURE 8-3
STRESSOR ENTER/EDIT OPTION
ROUTINE STRUCTURE

8.4.4 Variables

Only standard FileMan variables are used.

8.5 Clinical Data for Stressor Enter/Edit Option

8.5.1 Purpose

The Clinical Data for Stressor Enter/Edit option is used to enter data into the four fields that are related to the clinical aspects of exposure to the stressor: First Aid, Acute Effects, Chronic Effects, and Clinical Comments. The routine structure is shown in Figure 8-4.

8.5.2 Overview

The routine T2ESCL is used for the entry of clinical data into the Stressor file. This routine uses the standard Stressor file lookup routine T2SL to identify the stressor and DIE to control the data entry.

8.5.3 Globals Referenced

The following file is updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Stressor	---	1083	↑STRESS(0,	ADMIN	Update

8.5.4 Variables

Only standard FileMan variables are used.

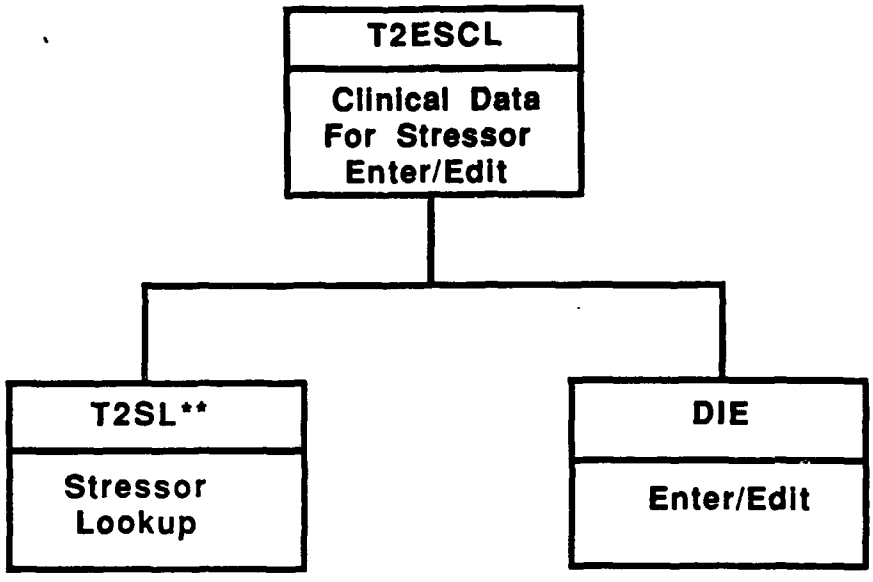

FIGURE 8-4
CLINICAL DATA FOR STRESSOR ENTER/EDIT OPTION ROUTINE STRUCTURE

9.0 UTILITIES

9.1 Introduction

The routines described in this section are utility routines. Each one may be called from any routine in any module.

9.2 Enter/Edit

9.2.1 Purpose

This utility is used to enhance the normal FileMan Enter/Edit process by ensuring that fields required for the prompt sequence are filled and that the entry passes specified consistency checks.

9.2.2 Overview

Routine T2GED performs this utility. The routine structure is illustrated in Figure 9-1. The routine calls routine DIE to execute a standard FileMan entry/edit prompt sequence. If the entry was deleted, control passes to line OK indicating that processing was completed properly. If there was a timeout on a new entry, control is passed to line DL which uses routine DIK to delete the entry before exiting. Otherwise, the entry is processed as specified by the GDR array for the file or subfile. First the file or subfile number is pulled from the (sub)file zero node. The routine then loops through the GDR array specified for the (sub)file and, through the semi-colon piece, the routine finds the location of the field information and pulls the field value from the file entry. If the field value passes the executed check, processing continues; otherwise, control is passed to line REQ.

After all the GDR array entries have been processed, if there are no multiple entry flags, the file entry is complete and control goes to line OK. If there is a multiple entry flag, line MUL is used to find whether any of the flagged entries within each specified multiple field exists. If any flagged entries exist, the entry fails the test and control is passed to line REQ; otherwise, the file entry does not have any existing incomplete multiples and control goes to line OK.

Starting at line REQ, the user is informed that the file entry has failed a check. If the entry is not new, control is passed back to line ED to try the edit again. For new entries, the user is asked whether the entry should be deleted or edited again. If an entry is to be deleted, line DL is used to call routine DIK, do the deletion, and exit; otherwise, control goes back to line ED.

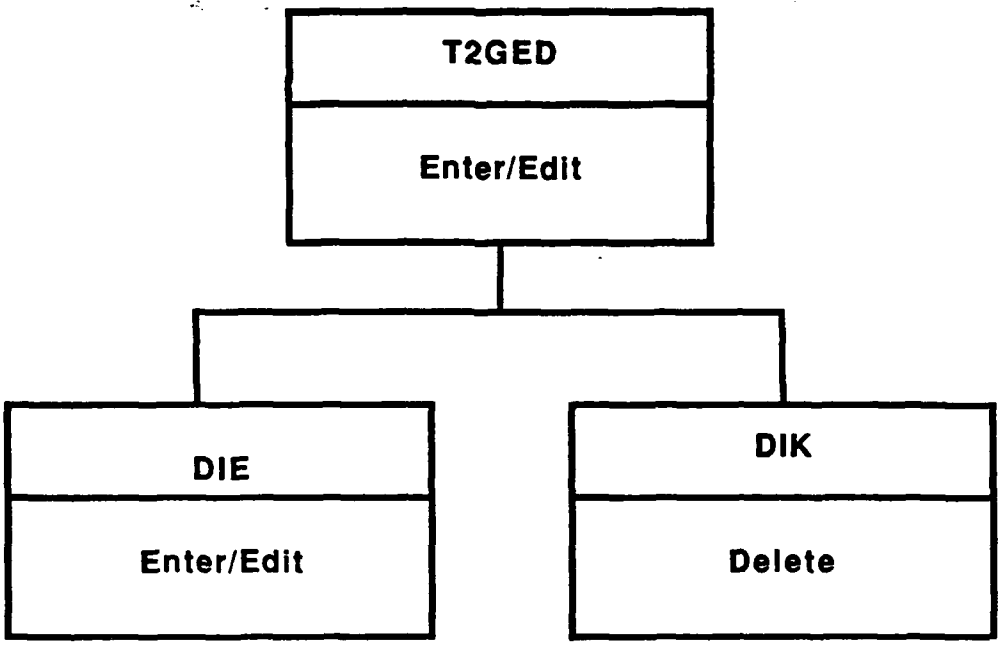

FIGURE 9-1
ENTER/EDIT UTILITY
ROUTINE STRUCTURE

9.2.3 Globals Referenced

Any user-specified file or subfile may be updated using this routine. The global reference will be in standard FileMan variable DIE.

9.2.4 Variables

Routine T2GED uses the following variables in addition to standard FileMan variables:

- F: The file or subfile number for the entry being processed.
- GDA: Typically this is the same as the Y value after a FileMan lookup. If the third piece is not zero, the entry being processed has just been added to the (sub)file and can be deleted; otherwise, the entry cannot be deleted.
- GDR: (1) An array containing information used to check the validity and consistency of a (sub)file number. The second subscript is a number which determines the order in which the array entries are processed.
  (2) The contents of a GDR array entry remaining to be processed.
- GFN: (1) The field number of the first field to be processed by routine DIE.
  (2) The field number being processed.
  (3) The field number for the field that failed a consistency or requirement check.
- GY: An array used to monitor the status of multiples within the (sub)file being processed. The first subscript is the multiple field number; the second is the entry number of a (sub)file entry in the multiple. Existence of an array entry indicates that the multiple entry was not fully processed or was deleted.
- I: The second subscript for a GDR array entry.
- J: A semi-colon piece of a GDR array entry; this contains either a field number of a field number followed by an ↑ and executable code.
- N: The node containing the field being processed.

- X: (1) A semi-colon piece of a GDR array entry.
  (2) The field number being processed.
  (3) The piece number that contains the field being processed.

- Y: (1) The entry node for the field being processed.
  (2) The field value of the field being processed.

9.2.5 Remarks

A field passes the requirement and consistency checks under the following circumstances:

- There is executable code in the GDR array entry for the field check which returns a true value when executed

- There is no executable code for the field check, the field is a multiple, and there is at lease one entry in the multiple subfile

- There is no executable code for the field check, the field is not a multiple, and the field value is not null

This routine will only function properly if the DR variable string or edit template is set up as described in the routine comment lines which are reproduced in Figure 9-1.

9.3 Employee Lookup

9.3.1 Purpose

This option looks up an entry in the Employee file and returns the entry number if the entry is found.

9.3.2 Overview

This lookup option, illustrated in Figure 9-2, calls routine T2PL. There are three entry points. Calling the routine at line ENT results in a prompt for employee. Entering at line EN1 bypasses the employee prompt and uses the lookup value in variable X. If JLIM has a value, a screen is set up to screen on agency. If PCOMP has a value, the screen also allows "Comp Only" employees to be selected. If the user has entered a space, DIC is called and will retrieve the most recently saved Employee file entry number. If the entry is in quotes, the routine is exited with a value of Y = -1. If the lookup value is preceded by "B/", it is presumed to be a badge number; if it is all numbers, it is presumed to be a social security number. Otherwise, the lookup is presumed to be on a name and routine T2PS is called to strip the name of any punctuation or spaces (see Section 9.12). The lookup is done against the pertinent cross reference using routine DIC at entry line IX.

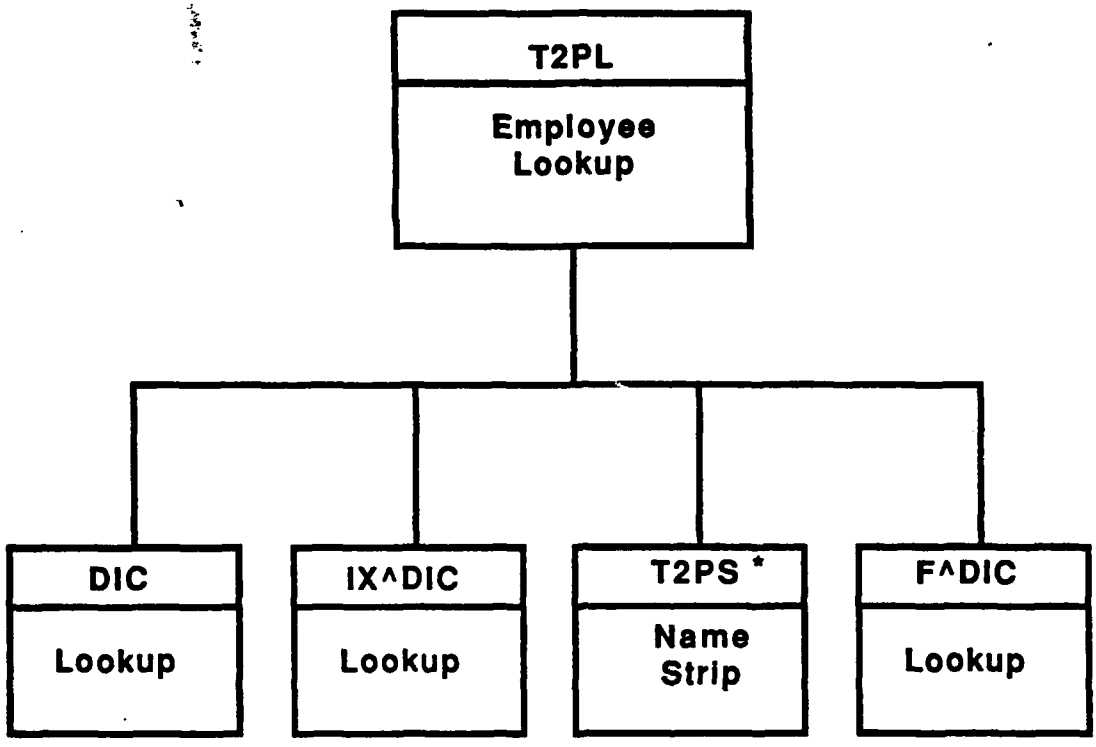

FIGURE 9-2
EMPLOYEE LOOKUP UTILITY
ROUTINE STRUCTURE

The EN2 entry point uses the same basic logic as the EN1 entry, except that the entry point used for routine DIC is line F not line IX. Entry at EN2 will bypass processing for space entries, which should have been handled by the DIC call, or when variable T2PL exists, indicating that routine T2PL has already been used to do the initial lookup.

9.3.3 Globals Referenced

The following global is read in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Employee	---	1004	↑EMPLOY(	ADMIN	Read

9.3.4 Variables

The following variables are used in this option:

- IN: The internal, stripped value of X (employee name)
- JLIM: If it exists, it contains the name of the agency unit to be used in the screen
- PCOMP: A flag to indicate that "Comp Only" employees are allowed in the lookup
- PH: The external value of X before being stripped of punctuation
- PNEW: The original value of X before it is stripped of spaces and punctuation
- T2PL: A flag to indicate that the routine has executed the instructions at entry point EN1 (i.e., performed the standard lookup)

9.3.5 Remarks

The EN2 entry point is needed to ensure that all Employee file lookups use the standardized social security number, badge, and employee name lookups. This entry point is called by data dictionary node ↑DD(1004,.01,7) when any lookup against the Employee file has not yet found a matching entry. This is needed for field lookups that point to

the Employee file where the pointing field is a multiple. Otherwise lookups done using DIC will not be sufficient to enable the desired capabilities, even if the ID field of the multiple calls routine T2PL in the input syntax check.

9.4 Stressor Lookup

9.4.1 Purpose

The special Stressor file lookup routine is used to enter a value that matches any of the cross referenced fields, and to enhance the display of candidate information for matches found in a key-word-in-context (KWIC) cross reference.

9.4.2 Overview

Routine T2SL is the special lookup routine for the Stressor file. Figure 9-3 shows the routine structure. There are two programmer entry points into the routine: ENT is used for lookups where the user should be prompted within the lookup, EN1 is used for lookups where the value on which the lookup is to be done (X variable) is already known. DIC is used to perform the lookup on the Stressor file.

The EN2 entry point uses the same basic logic as the EN1 entry, except that the entry point used for routine DIC is line F. Entry at EN2 will bypass processing for space entries, which should have been handled by the DIC call, or when variable T2PL exists, indicating that routine T2SL has already been used to do the initial lookup.

Lines ENTSYN through EX1 are executed out of the DIC("W") variable whenever a candidate list or single match is found. These lines are used to display the full synonym for instances when the displayed entry is a match on a synonym word found in the synonym KWIC cross reference, but not on the full synonym. Without this code, the user would see the matched synonym word and the file entry Primary Name, but not the full synonym.

9.4.3 Globals Referenced

The following global is read in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Stressor	---	1083	↑STRESS(0,	ADMIN	Read

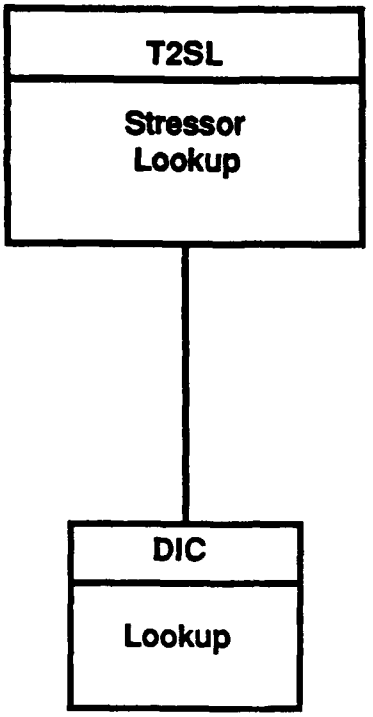

FIGURE 9-3
STRESSOR LOOKUP
ROUTINE STRUCTURE

9.4.4 Variables

The following variables are used in this routine:

- DD: FileMan variable
- DIX: FileMan variable
- DIY array: FileMan array
- DS array: FileMan array
- SA: Stressor file pointer for matched entry
- SB: Subfile pointer value for matched entry in cross reference
- SC: Text value of matched entry
- SD: Full name of Synonym field matched
- T2PL: Flag to indicate if the original DIC has been executed without finding a match

9.4.5 Remarks

The EN2 entry point is needed to ensure that all Stressor file lookups use the standardized lookup. This entry point is called by data dictionary node ↑DD(1083,.01,7) when any lookup against the Stressor file has not yet found a matching entry. This is needed for field lookups that point to the Stressor file where the pointing field is a multiple. Otherwise lookups done using DIC will not be sufficient to enable the desired capabilities, even if the ID field of the multiple calls routine T2SL in the input syntax check.

9.5 Agency Lookup

9.5.1 Purpose

The Agency Lookup routine prepares the appropriate screen logic (DIC("S")) for limiting the agency units found during a specific lookup, to subsequently invoke DIC, and to return the selected agency unit pointer value.

9.5.2 Overview

The routine T2JL is the special lookup routine for the Agency Unit file. The routine structure is illustrated in Figure 9-4. There are three entry points in this routine: EN0 prompts the user for a value

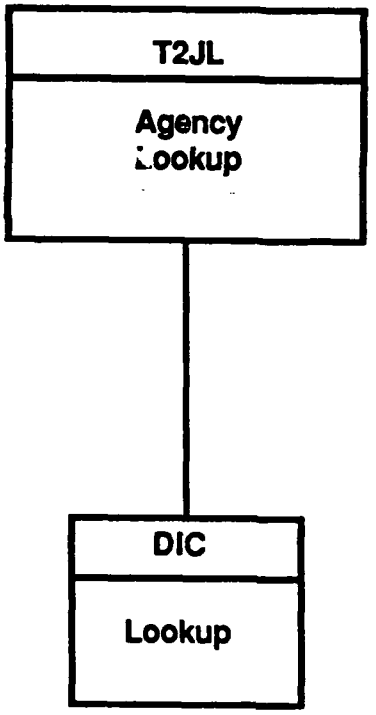

FIGURE 9-4
AGENCY LOOKUP UTILITY
ROUTINE STRUCTURE

before executing the lookup, EN1 is used when the lookup value is already known, and TOP is used when the lookup should be restricted to top-level agencies. In each case, DIC("S") is set to limit the matches found by DIC based on the value of variables that are set prior to executing this routine.

9.5.3 Globals Referenced

The following global is read in this routine:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read

9.5.4 Variables

The following variables are used in this routine:

- JAGU: Pointer value of agency unit
- JL: Flag to find inactive agency units (if $D)
- JLIM: Flag to limit the lookup to a specific agency
- JZZ: Flag to allow Prehire, Term, or Comp to be found

9.5.5 Remarks

This design idea was later superceded by choosing to make each lookup into the Agency Unit file set its own screen logic (DIC("S")) directly, rather than set up variables (flags) to communicate to T2JL. Some options in the system, however, still use T2JL.

9.6 Location Lookup

9.6.1 Purpose

This utility provides a standard lookup format for the Location file.

9.6.2 Overview

This utility is performed by routine T2GL. Figure 9-5 presents the routine structure. When used for prompting, this routine is entered at either line T2GL or ENT. This will save the existing screen, prompt for an entry and call entry point EN1 to do the lookup. If an entry is not selected, the routine will exit according to standard conventions or loop back to reprompt.

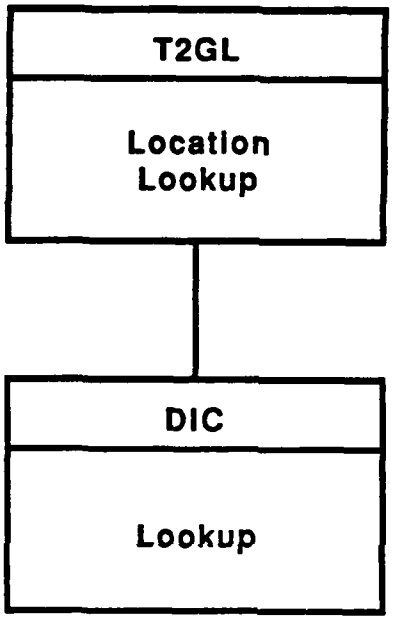

FIGURE 9-5
LOCATION LOOKUP UTILITY
ROUTINE STRUCTURE

Starting at entry point EN1, the user-entry is processed. An entry that does not meet basic syntax is immediately rejected. Otherwise, a screen is constructed using a furnished screen and the user entry. At the same time, the lookup value is changed and the cross reference to be used for the lookup is set based on the user entry. If the resulting lookup value is a question mark, the screen is killed and the standard response is displayed. In any case, the routine then calls routine DIC to look up on the selected cross reference. The routine is exited in the following situations: an entry is selected, the user entry specifies a wish to exit, adding a new entry is not allowed, the original entry contains an asterisk, or there is a matching entry that was not selected. Otherwise, the full user-entry is reconstructed and routine DIC is called to create a new entry.

9.6.3 Globals Referenced

The following globals are read and/or updated in this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Site	---	1041	↑AGENCY(1041,	ADMIN	Read
Location	---	1073	↑AGENCY(1073,	ADMIN	Update

9.6.4 Variables

Routine T2GL uses the following variables in addition to standard FileMan variables:

- GL: Identifies which comma piece of the user entry is used for the lookup
- GLAY: A non-zero value indicates that adding a new Location file entry is allowed
- GSS: The file screening value that was passed to the routine
- GX: (1) The user entered value
  (2) An array with the subscript from 1 to 4 and the contents being the corresponding comma piece value from the user-entry

9.6.5 Remarks

The lookup locates the last user-entry comma piece that is neither null nor an asterisk. This piece is used as the lookup value against the corresponding cross reference index.

When screening entries, the routine first uses any furnished screen. In addition, it will look for the following:

- File entries with a null in any piece for which the user entry consists of an asterisk
- File entries for which any piece corresponding to a user-entry piece that is not null or is all question marks, starts with the same characters as the user entry piece; in the case of the fourth piece, it can either start with or contain the character string from the user-entry piece

In effect, a null user-entry piece will allow selection of entries with any character string in that piece; an asterisk will select entries with a null in that piece; a final piece with question marks will provide a question mark response based on the selected cross reference; otherwise, the user-entry pieces limit selection to entries, starting with the furnished characters, or for the fourth piece containing the furnished characters.

9.7 Date/Time Utilities

9.7.1 Purpose

The date/time utility routines perform edit checking on values entered for fields that are to be a time of day. There are two different utilities; one checks times for military time format, the other conforms to the FileMan date/time rules for assuming AM and PM.

9.7.2 Overview

T2GTIMM is the routine used to check time data for military format and to convert the time entered into a 4 digit military time with leading zeros.

T2GTIM is the routine used to check time data that conforms to the rules for Fileman time data. Numbers between 6 and 11:59 are assumed to be for morning times (AM). Numbers between 12 and 5:59 are assumed to be for afternoons (PM). The routine returns a 4 digit military time with leading zeros.

9.7.3 Globals Referenced

There are no globals referenced by the time conversion/checking routines.

9.7.4 Variables

Only scratch variables are used in these routines.

9.8 Setup Agency Access String for User

9.8.1 Purpose

This option sets up a data variable for each user upon entry into the system. This variable limits the user access to data in specified agencies.

9.8.2 Overview

This option calls routine T2GUSER to set the DUZ("AG") variable equal to a string of agency codes that allows the user access to data for those agencies. If DUZ=0 (i.e., the user is a programmer or system manager) or the User file gives access to all agencies, then DUZ("AG") is set equal to a string of agency codes that includes all the codes in the Agency Unit file (taken from the "F" cross reference of that file). For all other users, the agency codes listed in the user file are used for the DUZ("AG") string.

9.8.3 Globals Referenced

The following globals are referenced for this option:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
User	---	3	↑DIC(3,	ADMIN	Read
Agency Unit	---	1074	↑AGENCY(0,	ADMIN	Read

9.8.4 Variables

The following variables are used in this option:

- GI: Scratch variable used as an index
- GJ: Agency code that is used as the entry number for the agency in the User file
- GK: Scratch variable used as an index
- GL: Agency code that is used as the entry number for the agency in the "F" cross reference of the Agency Unit file

9.9 Location Name Edit

9.9.1 Purpose

This utility checks the syntax of a location name to ensure validity.

9.9.2 Overview

Routine T2GLC performs this utility. No routines are called by routine T2GLC. First, the routine ensures that the first comma piece of the entry is not null, contains an existing site file entry, and is not being changed. Then each piece is checked to ensure it has a valid format. Finally, the routine checks to ensure there are no other identical entries that are not inactive.

9.9.3 Globals Referenced

The following globals are read:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Location	---	1073	↑AGENCY(1073,	ADMIN	Read
Site	---	1041	↑AGENCY(1041,	ADMIN	Read

9.9.4 Variables

The following variables are used:

- %GX: The user entry being processed
- GSI: The site portion of this Location file entry prior to the user entry being processed

9.10 Routine Tasking

9.10.1 Purpose

This utility allows tasking of routines using passed variables.

9.10.2 Overview

Routine T2GQTASK performs this utility. The routine structure is shown in Figure 9-6. After the routine establishes that the Task Manager is running, it calls routine %ZIS to select the tasking device, which is not to be opened at this point. If a spooler is chosen, routine %ZIS is called again to get the print device. In any case, the user is asked to verify the margin entered if it differs from the margin normally specified for the job. If the job is being run but is not being queued, the routine will pass control to the specified starting point. Otherwise, the routine establishes the time at which Task Manager is to start the job, files an entry in the tasking file including all existing variables and returns control to the specified exit point. If the user skips out of the prompt sequence at any point, control is passed to line X which tells the user to try later and passes control to the specified exit point.

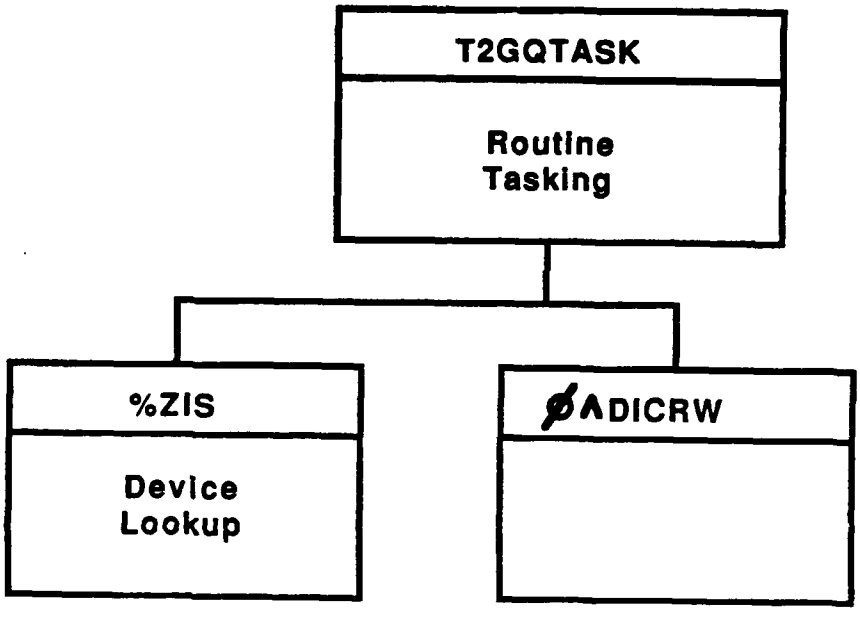

FIGURE 9-6
ROUTINE TASKING UTILITY
ROUTINE STRUCTURE

The tasker will pass control to this routine at the specified time, or as soon thereafter as the device is available, at line ZTSK. The routine then calls routine DICRW, reestablishes the saved variables, calls routine %ZIS to open the device with the proper parameters, and transfers control to the specified starting point.

The routine can be entered at line EX for use in closing the device, handling spooling, and executing special code before closing the device. For a spooling job, routine %ZIS is called to open the printing device.

9.10.3 Globals Referenced

The following globals are read and/or updated:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Device	---	3.5	↑%ZIS(1,	---	Read
Tasks	---	---	↑%ZTSK(	---	Update
Task Device	---	---	↑%ZTSCH(	---	Update

9.10.4 Variables

Routine T2GQTASK uses the following variables in addition to standard FileMan variables:

- GEX: The routine (and line) to which control is passed when a job is not to be run or has been tasked
- GIOEND: MUMPS code to be executed at the end of a run
- GIOM: The standard margin width for a job
- GO: The routine (and line) to which control is passed if a job is not tasked but is to be run
- GQTASK: Existence of this variable indicates that a device has not been selected
- GQTIME: Existence of this variable indicates that the job will be tasked to run immediately
- GQUE: Existence of this variable indicates that the job must be queued

9.10.5 Remarks

This routine was modified from FileMan routine DIO4 and should be kept as parallel with that routine as possible. Current uses do not

reenter this routine at line EX, but instead rely on forced tasking and the end of the task to close the device. As a consequence, any jobs tasked to the spooler would have to be printed by the operator from the spooler.

9.11 Occupation Medical/Training Requirements

9.11.1 Purpose

These utilities handle changes to medical and training requirements based on changes in an employee's occupation.

9.11.2 Overview

Routine T2PMOCC performs changes in the medical requirements. Figure 9-7 illustrates the routine structure. First, the routine establishes the Enrollment subfile zero node; if needed, it determines whether this is a new hire or a change in occupation and identifies the programs required by the occupation. The routine then processes each program for which the employee has an Enrollment subfile entry. If the employee is enrolled as required, the routine will warn the user about a program that is not required by the employee occupation. For programs that are required by the occupation, the routine selects the appropriate reexamination frequency and calls the Enrollment Update utility to modify an existing enrollment or add a new one. The Enrollment Update utility is also called to add enrollments in required programs for which the employee did not already have an Enrollment subfile entry.

Routine T2PTOCC is used to update the current course requirements for an employee, based on that employee's occupation. Figure 9-8 shows the routine structure. The routine builds a scratch table, TAB, which has for a subscript the entry number of required courses in the Course file. The Current Course subfile of the Employee file is updated to delete any courses not in the array, using routine DIE, and to add all the courses in the TAB array using routine DIC. The Course History subfile is searched for each required course, using routine DIC. If the required course is found in the Course History subfile, the qualifications date and status are added to the Current Course entry, using routine DIE.

9.11.3 Globals Referenced

The following globals are read and/or updated:

File Name	Subfile Name	File Number	Global Reference	Module Owner	Read or Update
Employee	Enrollment (Medical Program)	1004.05	↑EMPLOY(#,3,	ADMIN	Update
Occupation	---	1001	↑DIZ(1001,	ADMIN	Read

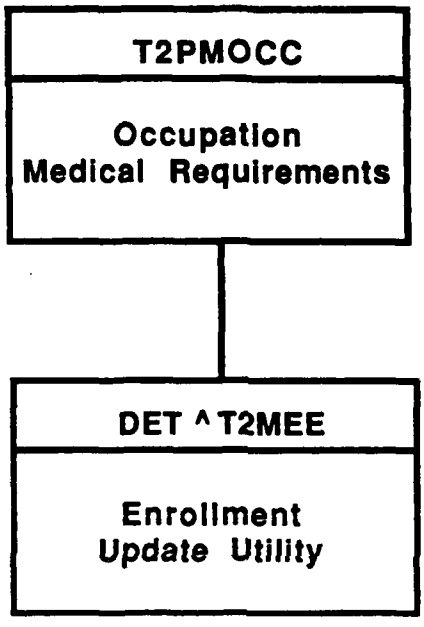

FIGURE 9-7
OCCUPATION MEDICAL REQUIREMENTS UTILITY
ROUTINE STRUCTURE

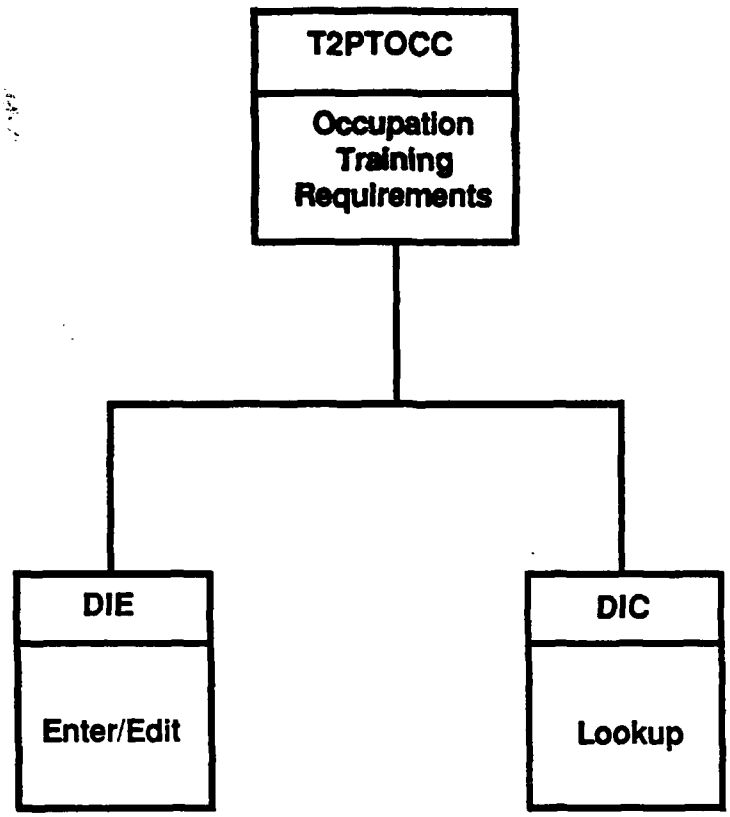

FIGURE 9-8
OCCUPATION TRAINING REQ UIREMENTS UTILITY ROUTINE STRUCTURE

Medical Program	---	1088	↑MED(1088,	MES	Read
Reason for Medical Visit/Exam	---	1128	↑MED(1128,	MES	Read
Employee	Current Course	1004.11	↑EMPLOY(#,10,	S/HT	Update
Employee	Course History	1004.12	↑EMPLOY(#,11,	S/HT	Update
Course Catalog	---	1113	↑TRN(1113,	S/HT	Read

9.11.4 Variables

Routine T2PMOCC uses the following variables in addition to standard FileMan variables:

- ME: A pointer to a Medical Program file entry and the corresponding employee Enrollment subfile entry number
- MF: The longest reexamination frequency for a program entry
- MME: A value used by the Enrollment Update utility to bypass user interaction
- MP: An Employee file entry number
- MPO: The zero node of a Medical Program file entry
- MR: An array using as subscripts the entry numbers of programs required for an occupation
- MRE: A variable set up and used by the Enrollment Update utility
- MRV: The Reason for Medical Visit/Exam entry appropriate to the employee action
- MYO: The zero node of an employee Enrollment subfile entry
- PON: The entry number of an employee's new occupation
- POO: The entry number of an employee's old occupation
- TAB: Temporary array whose subscript is the required course entry number
- TCOR: Course ID
- TDAT: Last qualification date

- TEFF: Date on which the course required event becomes effective
- TEMP: Entry number in Employee file
- TEN: Entry number of required course
- TMON: Number of months before retraining required
- TNEXT: Date retraining required
- TNUM: Entry number in Course History subfile
- TOCC: Entry number in Occupation file
- TQUAL: Last qualification date
- TREG: Date registered for course

9.11.5 Remarks

The Enrollment Update utility is discussed in Section 9.2 of the Medical Exam Scheduling Program Maintenance Manual.

9.12 Name Strip Utilities (T2PS, T2SSO)

9.12.1 Purpose

The Name Strip Utilities remove leading blanks and special characters from the names entered by the user. In the case of the Employee Name Strip Utility (T2PS), the routine strips extraneous characters for the user input and creates an Internal Name value if characters such as hyphen (-), apostrophe ('), semicolon (;), period (.), or blank are part of the Last Name. For the Stressor Name Strip Utility (T2SSO), the name is stripped of punctuation and numeric characters, for the creation of the Stripped Namel field.

9.12.2 Overview

The routine T2PS is the Name Strip Utility for the Employee Name. This code was based on the strip used in NOHIMS 1.0. The first action taken is to strip the value of spaces that precede or succeed any of the following characters: blank, comma (,), semicolon (;), hyphen (-), apostrophe ('). The leading and trailing blanks are removed from the value. This value is then replaced for the value the user entered. Then apostrophe (') or period (.) characters are removed. The Last Name is stripped of blanks and hyphen (-) characters. If the Last Name then does not match 1U.U, the routine returns a value of -1 in Y. At this point

attention is shifted to the First Name. The paragraph FN strips blanks from the First Name if it is not all uppercase, and allows a match on First Name. The paragraph FN strips blanks from the First Name. It allows all uppercase letters, a null First Name, or all uppercase letters followed by a comma and at least one number or uppercase letter. If the First Name has passed the check in paragraph FN, the variable IN is created from the stripped Last Name and the stripped First Name. If the First Name does not pass the check, a value of -1 is returned for Y. IN is set to null when it matches the value of the name after the initial removal of extraneous characters. If Y is less than 1, IN is set to null, so no Internal Name is returned from this routine.

The routine T2SS0 is the Name Strip Utility for the Stressor Name or Synonym. It is used by triggers on the Primary Name and Synonym fields of the Stressor file to create Stripped Names. Punctuation and numerics are stripped from the value, and multiple consecutive blanks are reduced to a single blank. Leading and trailing blanks are removed. If the value has not been stripped of punctuation or numeric characters, and it is only one word, the value returned is null. Otherwise, the value returned is the name after the stripping described above.

9.12.3 Globals Referenced

No globals are referenced within these utilities.

9.12.4 Variables

The variables used in the routine T2PS are the following:

- **%**: General scratch variable used as index variable of For loops and also as variable containing first name value for paragraph FN check
- **%(1)**: Index variable of For loop to contain each special character in turn for the initial strip
- **%(2)**: Index variable of For loop in initial strip of extraneous characters
- **%FNAM**: First Name after strip of extraneous characters
- **%IN**: Variable used for the initial strip of extraneous characters from the input value
- **%LNAM**: Last Name after strip of extraneous characters
- **%N**: Variable holding value for the action of paragraph STRIP

- X: Employee Name as entered by user, also the value of Employee Name after the initial strip of extraneous characters

- Y: Return variable; Y equals -1 when the name has been assessed as illegal as input

The variables used in T2SS0 are the following:

- A: Flag set to indicate whether punctuation or numeric characters have been removed from the input value

- X: Input value and value returned from the routine

10.0 UNDOCUMENTED FILEMAN FEATURES

10.1 Introduction

In the course of developing OSHRKS, several FileMan features were used that are not discussed in the standard FileMan documentation. This section provides documentation of those features.

10.2 File Lookup Options

There are several undocumented capabilities that are used during file lookups done using routine DIC. In addition to the documented parameters in variable DIC(0), this system uses two additional parameters. When a "Y" parameter is specified and there is more than one matching value, the system will return all matching values as subscripts in a Y array and the value of Y will be zero. A "U" parameter has FileMan do the lookup assuming the value given is in internal format rather than translating a given external value to an internal format, i.e., a pointer can be used instead of the external value, which may be ambiguous. This is useful in cases of pointers to the Employee file, where names will not be unique.

When the "U" parameter is used during lookups, the displayed entries list the .01 wi out translation, i.e., the pointer list, not the pointed-to value. In such cases, the display of the .01 field has been suppressed by setting variable DICRS. This variable must be killed after the lookup is completed.

In some instances where an entry can be added to a file, the variable DIC("DR") has been set. This variable contains a DR prompt sequence that overrides the typical identifier prompts. The lookup must be done with parameter "E" for this variable to have any effect.

Instead of using a DIC("W") variable to change the entry display during lookup, a standard lookup display has been established for several files by setting nodes under ↑DD(1004,0,"ID") in the data dictionary. The fourth subscript in this array is usually an identifier field number, but an alpha subscript allows a string instead of, or in addition to, the usual identifiers. Since these array members are used to construct a DIC("W") variable, there are system restrictions that limit the total array length. Also, the subfile or file zero node must indicate that there is an identifier.

10.3 Subfile Lookups and Edits

In many cases, subfile lookups and adds are done directly by DIC rather than having FileMan do the lookup for a multiple field in an edit

mode. To do a direct lookup, the subfile zero node must exist before the call to DIC. This node contains the file number and special lookup parameters which control lookup prompts and displays and describe the .01 field variable type. An "A" parameter indicates that the user should be prompted before adding a new value. An "0" parameter indicates that the user should be asked if a single found entry is "OK?" An "I" parameter indicates that there are identifier fields. In addition to requiring the zero node, a subfile lookup must have the full global reference for the subfile in variable DIC and must have a DA array ranging from DA(n) to DA(1) where DA(n) is the file entry number and DA(1) is the subfile entry number for the subfile above the one being looked up.

For a subfile edit, there must be a subfile zero node, variable DIE must contain the subfile global reference, and there must be a DA array as described for the subfile lookup and a variable DA containing the entry number being processed.

10.4 Entry Number Determination

In several cases, the variable DINUM has been used to determine the entry number for a file or subfile entry. MUMPS code which sets DINUM can be found in the input syntax check code for a file or subfile .01 field. Most often this was used to file dates in inverted order (to facilitate their retrieval) or to make sure a date or pointer only has one entry per value. When the variable DINUM is in the input syntax, the system will not allow the user to change the .01 value for an entry.

10.5 Trigger Protection

To protect file triggers from deletion, the node ↑DD(#,FN,1,TN,3) can be set, where # is the (sub)file number, FN is the field number, and TN is the cross reference number for the trigger. Protections were established on triggers, using special MUMPS code that cannot be created using standard FileMan.

10.6 Limits on Entry Deletion

For some files or subfiles, node ↑DD(#,01,"DEL",1,0), where # is the (sub)file number, contains MUMPS code that limits or prohibits deletion of entries from the file or subfile. If a non-zero MUMPS truth value is returned when this code is executed, an entry cannot be deleted by any user. This was used to protect many of the standard reference files which are pointed to by other files. This feature can be disabled by preceding the node with "I 0;". This should only be disabled if essential during installation before a file entry is used by another file. The original code must be restored or system integrity will be compromised.

10.7 Entry/Edit Value Stuffing

The FileMan documentation discusses the use of three slashes ("///") to force a field value. This format still executes the input syntax code and will fail if there are ambiguities. Particularly for pointers, the OSHRKS system often uses a four slash stuff, which tells FileMan to file the provided value exactly as is without doing any input syntax or translation checking.

APPENDIX A
CROSS REFERENCE OF OPTION NAMES TO OPTIONS

OPTION TEXT	OPTION NAME
Agency Edit	T2J Edit Agency
Agency Functions	T2J Agency Menu
Agency Outline List	T2J Agency Indented List
Agency Units by Level	T2J Agency Units by Level
Agency Units by Site	T2J Agency Units by Site
Assign Agency Access to Users	T2G Assign User Access
Assign Employee to Location	T2P Assign Emp to Loc
Assign Location for Agency Unit	T2J Assign Loc to Agency Unit
Clinic Table Enter/Edit	T2MCE Clinic Entry
Clinical Data for Stressor Enter/Edit	T2E Stressor Clinical Data
Create New Agency	T2J Create New Agency
Delete NCPDS Transactions	T2PUTD Delete Transactions
Display Employee	T2G Display Employee
Edit NCPDS Transaction File	T2PUTE NCPDS Transaction Edit
Edit Operation Class Name	T2G Set Ops Class
Edit Operation Subclass Name	T2G Set Ops Subclass
Enter/Edit Compensation Only Employee	T2P Comp Only Enter/Edit
Enter/Edit Employee	T2P Employee Enter/Edit
Enter/Edit Location	T2G Location Enter/Edit
General System Tables	T2G General Tables
Inactivate Agency Unit	T2J Inactivate Agency Unit
Inactivate Location	T2G Location Inactivate
Inquiry for Agency Unit	T2J Agency Inquiry
List Employees by Agency Unit	T2P Employees by Agency Unit
List Employees by Location	T2P Emp by Location
Load NCPDS Tape into Transactions File	T2PUTL NCPDS Tape Load
Location Functions	T2G Location Options
Module Tables and Files	T2G Module Tables
NCPDS Functions	T2PU NCPDS Options
Personnel Functions	T2P Personnel Options
Print Transactions	T2PUTP Print Transactions
Sample Units Enter/Edit	T2E Setup Sample Units
Search and Print File Entries	T2G File Reports
Set Up Deficiency Tables	T2G Deficiency
Set Up Environmental Tables	T2G Environ
Set Up Medical Tables	T2G Medical
Set Up Occupation File	T2G Setup Occupation
Set Up Operations File	T2G Setup Operations
Set Up Organization Levels	T2J Setup Org Levels
Set Up Site file	T2G Setup Site
Stressor Class Enter/Edit	T2E Setup Stressor Class
Stressor File Enter/Edit	T2G Setup Stressor File
Terminate Employee	T2P Terminate Employee
Transfer Employee (Shop)	T2P Transfer Employee
Update Employee file from NCPDS Transactions	T2PUET Update Employee File

APPENDIX B
CROSS REFERENCE OF PRINT TEMPLATES TO OPTIONS

PRINT TEMPLATE	OPTION
Agency Unit	Agency Units by Level
	Agency Units by Site
	Inquiry for Agency Unit
Emp By Agency	List Employees by Agency Unit
Emp By Location	List Employees by Location
Employee Display	Display Employee
Tape Transactions	Print Transactions

APPENDIX C
CROSS REFERENCE OF SORT TEMPLATES TO OPTIONS

SORT TEMPLATE	OPTION
Agency Level	Agency Units by Level
Agency Site	Agency Units by Site
Current Emp in Agency Unit	List Employees by Agency Unit
Special Employees	List Employees by Agency Unit
T2EL	List Employees by Location

APPENDIX D
CROSS REFERENCE OF ROUTINE ENTRY POINTS TO OPTIONS

ROUTINE ENTRY POINT	OPTION
↑%DT	Delete NCPDS Transactions
	Load NCPDS Transaction File
	Print Transactions
	Update Employee File from NCPDS Transactions
↑%ZIS	Update Employee File from NCPDS Transactions
	Routine Tasking Utility
↑DIC	Agency Edit
	Agency Lookup Utility
	Agency Units by Level
	Agency Units by Site
	Assign Agency Access to Users
	Assign Location for Agency Unit
	Create New Agency
	Delete NCPDS Transactions
	Edit NCPDS Transaction File
	Edit Operation Class Name
	Edit Operation Subclass Name
	Employee Lookup Utility
	Enter/Edit Employee
	Enter/Edit Location
	Inactivate Agency Unit
	Inactivate Location
	Inquiry for Agency Unit
	List Employees by Agency Unit
	Load NCPDS Transaction File
	Location Lookup Utility
	Occupation Training Requirements Utility
	Print Transactions
	Sample Units Enter/Edit
	Set Up Occupation File
	Set Up Operations File
	Set Up Site File
	Stressor Class Enter/Edit
	Stressor Lookup Utility
	Update Employee File from NCPDS Transactions
F↑DIC	Employee Lookup Utility
	Stressor Lookup Utility
IX↑DIC	Agency Units by Level
	Employee Lookup Utility

ROUTINE ENTRY POINT	OPTION
WAIT↑DICD	Agency Edit
	Delete NCPDS Transactions
	Enter/Edit Employee
	Terminate Employee
O↑DICRW	Routine Tasking Utility
↑DIE	Agency Edit
	Assign Agency Access to Users
	Assign Employee to Location
	Assign Location for Agency Unit
	Clinical Data for Stressor Enter/Edit
	Create New Agency
	Edit Operation Class/Name
	Edit Operation Subclass Name
	Enter/Edit Location
	Enter/Edit Utility
	Inactivate Agency Unit
	Load NCPDS Transaction File
	Occupation Training Requirements Utility
	Stressor Enter/Edit
	Terminate Employee
	Update Employee File from NCPDS Transactions
↑DIK	Create New Agency
	Delete NCPDS Transactions
	Enter/Edit Compensation Only Employee
	Enter/Edit Employee
	Enter/Edit Utility
	Load NCPDS Transaction File
	Terminate Employee
	Update Employee File from NCPDS Transactions
EN1↑DIP	Agency Units by Level
	Agency Units by Site
	Display Employee
	Inquiry for Agency Unit
	List Employees by Agency Unit
	List Employees by Location
	Print Transactions
EN↑DIQ	Agency Edit
	Assign Location for Agency Unit
↑T2ESCL	Clinical Data for Stressor Enter/Edit

ROUTINE ENTRY POINT	OPTION
↑T2ETSTR	Stressor Class Enter Edit
↑T2ETUNT	Sample Units Enter/Edit
↑T2GED	Enter/Edit Utility Agency Edit Assign Location for Agency Unit Create New Agency Edit NCPDS Transaction File Enter/Edit Compensation Only Employee Enter/Edit Employee Inactivate Agency Unit Inactivate Location Sample Units Enter/Edit Set Up Occupation File Set Up Operations File Set Up Organization Levels Set Up Site File Stressor Class Enter/Edit Transfer Employee (Shop)
↑T2GL	Location Lookup Utility Assign Employee to Location Assign Location for Agency Unit Enter/Edit Location Inactivate Location
EN1↑T2GL	Location Lookup Utility List Employees by Location
↑T2GLC	Location Name Edit Utility
↑T2GLE	Enter/Edit Location
ENT↑T2GLI	Inactivate Location
↑T2GQTASK	Routine Tasking Utility
↑T2GTIM	Date/Time Utility
↑T2GTIMM	Date Time Utility
ENT↑T2GTOCC	Set Up Occupation File
CLASS↑T2GTOPR	Edit Operation Class Name

ROUTINE ENTRY POINT	OPTION
ENT↑T2GTOPR	Setup Operations File
OPSUB↑T2GTOPR	Edit Operation Subclass Name
↑T2GTSIT	Set Up Site File
↑T2GUSEN	Assign Agency Access to Users
↑T2GUSER	Set Up Agency Access String for User Utility
ENT↑T2JA	Assign Location for Agency Unit
↑T2JEN	Agency Edit
EDIT↑T2JEN	Create New Agency
ENT↑T2JEN1	Agency Edit
HF↑T2JEN1	Agency Edit Create New Agency
↑T2JENL	Create New Agency
ENT↑T2JI	Inactivate Agency Unit
↑T2JL	Agency Lookup
TOP↑T2JL	Agency Lookup
↑T2JLMF	Agency Outline List
↑T2JLH	Assign Location for Agency Unit
↑T2JR1	Inquiry for Agency Unit
↑T2JR2	Agency Outline List
↑T2JRL	Agency Units by Level
↑T2JRS	Agency Units by Site
↑T2JTLEV	Set Up Organization Levels
DET↑T2MEE	Occupation Medical Requirements Utility

ROUTINE ENTRY POINT	OPTION
DET↑T2MER	Terminate Employee
ENT↑T2PA	Assign Employee to Location
↑T2PEN	Enter/Edit Employee
CEDIT↑T2PEN	Enter/Edit Compensation Only Employee
COMP↑T2PEN	Enter/Edit Compensation Only Employee
ENT↑T2PEN	Enter/Edit Compensation Only Employee Enter/Edit Employee
ENTC↑T2PEN	Enter/Edit Compensation Only Employee
OCC↑T2PEN	Update Employee File from NCPDS Transactions
LOC↑T2PH	Assign Employee to Location Terminate Employee
↑T2PL	Employee Lookup Utility Assign Employee to Location Display Employee Enter/Edit Employee Terminate Employee Transfer Employee (Shop)
EN1↑T2PL	Employee Lookup Utility
EN2↑T2PL	Employee Lookup Utility
↑T2PMOCC	Occupation Medical Requirements Utility Enter/Edit Employee
↑T2PR2	List Employees by Agency Units
↑T2PR3	Display Employee
↑T2PR5	List Employees by Location
↑T2PS	Name Strip Utility Employee Lookup Utility
↑T2PTERM	Terminate Employee Update Employee File from NCPDS Transactions

ROUTINE ENTRY POINT	OPTION
↑T2PTOCC	Occupation Training Requirements Utility Enter/Edit Employee
ENT↑T2PTRA	Transfer Employee (Shop)
↑T2PUET	Update Employee File from NCPDS Transactions
↑T2PUTD	Delete NCPDS Transactions
↑T2PUTE	Edit NCPDS Transaction File
↑T2PUTL	Load NCPDS Transaction File
↑T2PUTP	Print Transactions
↑T2SEN	Stressor Enter/Edit
↑T2SL	Stressor Lookup Utility Clinical Data for Stressor Enter/Edit Stressor Enter/Edit
EN1↑T2SL	Stressor Lookup Utility
EN2↑T2SL	Stressor Lookup Utility
↑T2SSO	Name Strip Utility